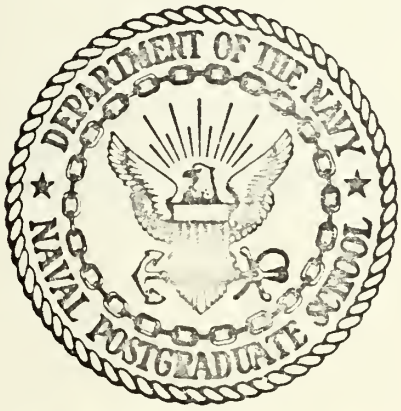
A COMPUTERIZED ALGORITHM FOR SEQUENTIAL SEARCH OF THE GLOBAL MAXIMUM

by

Ray Lovell Springfield

United States
Naval Postgraduate School



THESIS

A Computerized Algorithm for Sequential

Search of the Global Maximum

by

Ray Lovell Springfield

September 1970

A Computerized Algorithm for Sequential Search of the Global Maximum

by

Ray Lovell Springfield
Captain, United States Marine Corps Reserve
B.S. , Mansfield State College, 1963

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 1970

ABSTRACT

A sequential search procedure for maximization of a single variable multimodal objective function is designed and investigated in this research. Existing sequential procedures require the function to be unimodal. Nonsequential methods, though not restricted in this sense, require a large number of samples. Results show that the proposed sequential method is in this case preferable.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

25. Consequences of measurement errors.

26. Case where smaller C than required is not recognized by data set.

27. Case where C is underestimated but has no effect on the algorithm.

28. Illustration of "black box" case.

I. INTRODUCTION

A. PURPOSE

The purpose of this research is to design a sequential search procedure capable of estimating and locating the global maximum of a single variable, multimodal objective function with a predetermined accuracy. The procedure will be derived in the form of an algorithm such that it can be easily implemented by a digital computer.

B. LITERATURE SURVEY

A maximization problem for a digital computer provides an algorithm by which the objective function is evaluated or sampled for various values of its argument (sampling points). Such a procedure must prescribe a rule to choose the sampling points (to be called the sample rule) and a function (to be called the estimate) depending, in general, on all samples obtained and approximating the unknown value of the maximum, Shubert and Spang [Refs. 10 and 11]. The sampling rule divides these procedures into two general categories: sequential and nonsequential. In a sequential procedure the sampling rule utilizes the values of previous samples to determine the location of the next sampling point. The procedure is called nonsequential if the sampling points are chosen, possibly by some random mechanism, in advance of any computation or experimentation.

Hill, Spang and Wilde [Refs. 4, 11, and 13] have further subdivided these procedures into three basic groups: (1) gradient methods; (2) sequential min-max methods; and (3) random and grid search methods.

1. Gradient Methods

The largest body of material in the literature is concerned with what has been referred to as "gradient methods" of maximization. These methods utilize the "hill-climbing" principle to determine the direction in which the objective function increases, i.e., measurements of the slope of the function are used as an indication of the direction toward the maximum. A general approach to gradient methods is found in Spang [Ref. 11]. The first sampling point is selected arbitrarily and depends primarily on the experimenter's subjective opinion as to the location of the maximum. If the objective function is such that the approximate location of the maximum cannot be determined in a subjective manner, the midpoint of the experimental region may be

used as the "initial guess", Brooks [Ref. 2], where the experimental region is defined as the interval containing all possible sampling points for which the objective function is defined. Computing time will be significantly reduced the closer the initial sampling point is to the true abscissa of the maximum. The remaining sampling points are determined by the iterative equation:

$$x_{n+1} = x_n + h_n D_n \tag{1}$$

where x_n is the value of the sampling point at the n th iteration, h_n is a positive constant, and D_n is the direction vector evaluated at the n th iteration. For a single variable objective function, D_n is, of course, a scalar. Thus, if D_n is positive the (n + 1) st sampling point will be to the right of the n th sampling point, the reverse will hold if D_n is negative. The magnitude of $h_n D_n$ determines how large a step is taken in the direction specified by D_n. The iterations continue until

$$h_n D_n \leq \epsilon \tag{2}$$

where $\epsilon > 0$ is the desired accuracy in estimating the true location of the maximum. The inequality given by (2) is considered the stopping rule for this method. If the inequality is satisfied then x_n is the estimate for the abscissa of the maximum and $f(x_n)$ becomes the estimate for the maximum. If the inequality is not satisfied, then the procedure goes to (1). Although the various gradient methods differ in their choice of scale factor h_n and direction vector D_n, the general approach to the problem is the same.

For example, Spang [Ref. 11] uses as a choice of the direction vector D_n

$$\left[\frac{df}{dx}\right]_n , \tag{3}$$

where (3) is the value of the derivative at the nth sampling point. The sample values of the (n + 1) st and nth sampling points are used to choose the value of h_{n+1} in the following manner:

$$f(x_{n+1}) > f(x_n) \text{ then } h_{n+1} = h_n$$

$$f(x_{n+1}) \leq f(x_n) \text{ then } h_{n+1} = \frac{h_n}{2}$$

2. Sequential Min-Max Methods

Like gradient methods, sequential min-max methods operate under the assumption that the objective function is unimodal in the experimental region. In general, these procedures reduce the range of the independent variable by a predetermined amount. Hence, it is possible to determine, before taking any samples, the number of iterations required to reduce the range of the independent variable a given amount. The method developed by Kiefer [Ref. 7], based on the Fibonacci numbers, is the most popular of all min-max methods. No other method can guarantee a shorter interval of uncertainty for all functions of the class considered (unimodal functions defined in the experimental region [a,b].) Another related method can be found in Berman [Ref. 1]. Spang and Wilde [Refs. 11 and 13] outline the general approach discovered by Kiefer.

This approach assumes that the maximum of the objective function initially lies within an interval a_n, b_n as illustrated by Fig. 1. If two sampling points are selected within this interval, say x^n_1 and x^n_2, where n is the iteration number and such that $x^n_1 < x^n_2$, it is obvious that if

$f(x^n_1) > f(x^n_2)$ then the maximum lies between a_n, x^n_2

$f(x^n_1) < f x^n_2)$ then the maximum lies between x^n_2, b_n

$f(x^n_1) = f(x^n_2)$ then the maximum lies between x^n_1, x^n_2.

Whenever the equality condition occurs, either the interval $[a_n, x^n_2]$ or $[x^n_2, b_h]$ is selected to maintain mathematical symmetry. Thus, by this simple test the size of the interval guaranteed to contain the maximum can be reduced.

The sampling rule is based on the total number of tests, N, that must be performed. N can be determined once the experimenter has specified the desired accuracy in estimating the true location of the maximum. The length of the final interval, $b_N - a_N = \delta_N$ specifies the desired accuracy.

Kiefer [Ref. 7] has shown that after N iterations

$$\delta_N = \frac{1}{2U_N}(b_0 - a_0), \tag{4}$$

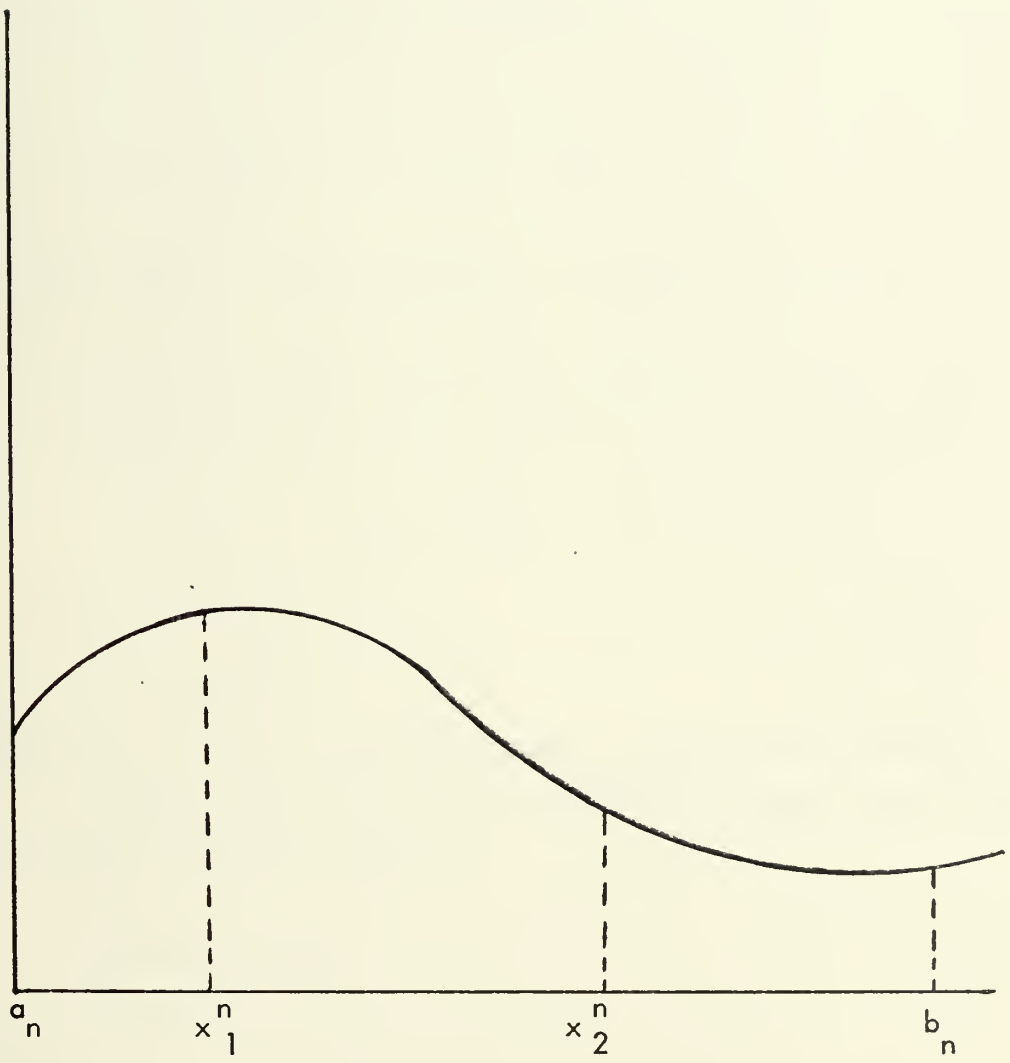


Figure 1. Illustration for sequential min-max method

where U_n is the value of the N th Fibonacci number and $(b_0 - a_0)$ is the length of the original interval, see also Spang [Ref. 11]. Solving (4) for U_N, the value of the N th Fibonacci number is determined. Using the table of Fibonacci numbers, the corresponding N, which is the total number of tests required to attain the desired accuracy δ_N, can be found.

With N established the algorithm proceeds in the following manner:

1. $$x_1^n = \frac{U_{N-1-n}}{U_{N+1-n}} (b_n - a_n) + a_n$$

$$x_2^n = \frac{U_{N-n}}{U_{N+1-n}} (b_n - a_n) + a_n$$

2. $f(x_1^n) > f(x_2^n)$ set $a_{n+1} = a_n$, $b_{n+1} = x_2^n$

$f(x_1^n) < f(x_2^n)$ set $a_{n+1} = x_2^n$, $b_{n+1} = b_n$

$f(x_1^n) = f(x_2^n)$ set $a_{n+1} = a_n$, $b_{n+1} = x_2^n$

or set $a_{n+1} = x_1^n$, $b_{n+1} = b_n$.

3. $n = N$, set $\delta_N = b_n - a_n$

$n \neq N$, go to 1.

Either $f(a_N)$ or $f(b_N)$ could be used as an estimate of the maximum.

For large values of n the ratios of the Fibonacci numbers given in step 1 of the algorithm approach a constant:

$$\frac{U_{n-1}}{U_{n+1}} \approx 0.382$$

and

$$\frac{U_n}{U_{n+1}} \approx 0.618$$

Therefore the following approximation formulas can be used to obtain x_1^n and x_2^n, for n = 1, . . . , N:

$$x_1^n = 0.382\,(b_n - a_n) + a_n$$

$$x_2^n = 0.618\,(b_n - a_n) + a_n,$$

see Spang [Ref. 11].

3. Disadvantages of Gradient and Sequential Min-Max Methods

The major drawback of these methods is that they are successful only if the objective function is unimodal, i.e., has only one hump in the experimental region. If the objective function does not satisfy the unimodality requirement, these methods will be successful in reaching a local maximum at the best. This is obvious because the methods are based on the "hill-climbing" principle of moving the next sampling points in the direction in which the function increases. Since in many practical problems it is not possible to guarantee unimodality of the objective function, it is important to develop a search technique which finds the maximum of a multimodal objective function. Furthermore, gradient methods usually require further regularity conditions such as the existence of the first and second derivatives.

4. Random Search Methods

Unlike gradient and sequential min-max procedures, random search procedures are not restricted to unimodal functions. These methods are nonsequential in that the previous sample values do not determine exactly where the next sampling point will be located. Various purely random methods can be found in Brooks, Karnopp, Spang and Zachharov [Refs. 3, 6, 11, and 14].

The general procedure is to select the sampling points at random in the area where the maximum is located according to a fixed distribution. After a certain number of iterations have been performed, the largest value of the objective function is considered to be the estimate of the maximum. Assuming that the maximum is equally likely to occur anywhere in the experimental region, [a, b], let (b - a) = d be the length of this interval. With no prior knowledge concerning the location of the maximum, it is reasonable to use a flat density function over the interval [a, b]. A priori, the experimenter specifies the accuracy he desires in estimating the location of the maximum as δ_N, where δ_N is the largest

interval of uncertainty he is willing to accept after p iterations. The interval [a, b] is further divided into N subintervals each of length δ_N. The ratio of a subinterval to the original interval is

$$g = \frac{\delta_N}{d} .$$

The probability that a sampling point is not in a particular interval is (1 - g) and the probability that it is still not in this interval after p trials is $(1 - g)^p$. Thus, the probability of at least one of the sampling points being in this subinterval is

$$s = 1 - (1 - g)^p . \tag{5}$$

s can also be considered as the confidence level of one of the sampling points being in a specified interval. Solving (5) for p, the required number of sampling points can be determined as

$$p = \frac{\log (1 - s)}{\log (1 - g)} . \tag{6}$$

Brooks [Ref. 3] and Spang [Ref. 11] tabulate the number of iterations required for various values of g and s. It can be seen from these tables that the number of sampling points (iterations) required increase quite rapidly with a reduction in g.

Suppose x_i is the value of the sampling point at the i th iteration, where i runs from 1 to p, p determined by (6); R_i is the value of a random number between 0 and 1 selected from a uniform distribution for the i th iteration; and $\hat{f}_i$ is the estimate of the global maximum at the i th iteration; then the algorithm follows:

1. Set i = 1
2. Compute
   $x_1 = a + R_1 d$
   $\hat{f}_1 = f (x_1)$
3. Set i = i + 1
   Compute
   $x_i = a + R_i d$
   $\hat{f}_i = \max \{ \hat{f}_{i-1}, f (x_i) \}$

4. $i = p$, estimate the global maximum to be $\hat{f}_p$ at x_p.

   $i \neq p$, go to 3.

5. Grid Search Methods

Grid search procedures are systematic in the sense that the sampling points are equally spaced a predetermined distance apart in the experimental region. The sample values for all sampling points are obtained and that which is the largest is considered the estimate of the maximum.

Like random procedures and unlike gradient and sequential min-max procedures, these methods are successful in estimating the global maximum and its location for a multimodal objective function over the experimental region. These methods are also nonsequential. A method for grid search can be found in Spang [Ref. 11].

In general, the experimental region [a, b] is subdivided into N intervals of length δ_N, where δ_N is the accuracy the experimenter is willing to accept in estimating the location of the global maximum for the objective function in [a, b]. The number of sampling points, p, required by this procedure can easily be computed as

$$p = \frac{d}{\delta_N} , \tag{7}$$

where d is the length of the original interval. This is about half as many iterations as are required by purely random methods to attain the same accuracy.

Suppose x_i is the value of the sampling point at the i th iteration, where i runs from 1 to p, p determined by (7); $\frac{d}{\delta_N}$ is the length of the equidistant intervals; and $\hat{f}_i$ is the estimate of the global maximum at the i th iteration; then the algorithm follows:

1. Set $i = 1$
2. Compute

$$x_1 = a + \frac{d}{\delta_N}$$

$$\hat{f}_1 = f(x_1)$$

3. Set $i = i + 1$

Compute

$$x_i = x_{i-1} + \frac{d}{\delta N}$$

$$\hat{f}_i = \max \{ \hat{f}_{i-1}, f(x_i) \}$$

4. $i = p$, estimate global maximum to be $\hat{f}_p$ at x_p.

$i \neq p$, go to 3.

6. Disadvantages of Random and Grid Search Methods

The major drawback to random procedures is that the maximum is found only with some probability as long as the number of samples is finite. Furthermore, being nonsequential, random methods as well as grid search procedures require a very large number of samples to estimate the maximum and its location with reasonable confidence level and residual uncertainty. Although grid search procedures require about half the number of sample points required by random methods to attain the same accuracy, the number of samples required is still too large to be practical in many situations.

7. Methods Combining Gradient and Nonsequential Search Procedures

Several attempts have been made to combine the "hill-climbing" principle with nonsequential search to maximize a multimodal function over the experimental region. For various methods utilizing these procedures see Hill, Hartley, Matyas, Pijavskii, Vaysbord and Yudin [Ref. 4, 5, 8, 9, and 12].

Basically two approaches are used in this type of search: (1) finite random or deterministic global search procedures are used to locate favorable starting points in the experimental region with gradient methods applied in the intervals specified by the starting points; and (2) gradient methods are applied in the current interval being searched and at some random time during the search of this interval the search goes to another randomly selected interval in the experimental region; Matyas, and Vaysbord and Yudin [Refs. 8 and 12] illustrate this approach.

In approach (1) the experimental region [a, b] is divided into N subintervals by some finite random or deterministic method. Each of the intervals specified by this procedure are then searched by gradient methods. The largest

sample value obtained is the estimate of the global maximum. The drawback of this method is that it does not guarantee that the global maximum will eventually be found.

Approach (2) uses gradient methods about the initial sampling point until a random trial moves the search to another interval. After each sample is observed this random trial is performed. If the procedure moves to another interval, still another random trial is used to determine the exact interval to be searched. The method continues until the stopping rule is satisfied at which time the estimate for the global maximum is considered to be the highest sample value attained. The drawback of this method is that the disadvantages of purely random methods prevail, namely, a very large number of iterations (samples) are required.

8. Conclusions

The methods discussed above are either too stringent on regularity conditions and the requirement that the function be unimodal in the experimental region or impractical from the standpoint of the number of iterations required. Furthermore, none of these methods provide a truly sequential approach to the problem of multimodality.

C. APPROACH OF THE METHOD TO BE CONSIDERED

In this research the approach will be to solve the maximization problem of multimodality in terms of a sequential sampling rule which is easily implemented. The formulation will restrict the class of admissable functions to be maximized to those of a single variable and which are globally Lipschitzian. In addition it will be assumed that there are never any errors present in the observed values of the function. The assumption that the function be globally Lipschitzian means that there is some constant C, the value of which is known, such that

$$C \geq \frac{| f(x) - f(x') |}{| x - x' |} \tag{8}$$

for any x, x', $x \neq x'$, in the experimental region $[a, b]$. If the function is differentiable, the value of C is usually not too difficult to compute. It amounts to finding some upperbound on the function's derivative. In the case where the function is given empirically, the constant C can often be obtained from the

physical nature of the function. If the exact form of the function is unknown, the selection of C will have to be based on the experimenter's subjective judgement. In any case, if it is desired to estimate the value or location of the maximum with a predetermined accuracy, knowledge of C or some equivalent information is necessary to determine a stopping rule regardless of the method used. This is obvious since if nothing of this sort is known or assumed about the function, no conclusion can be drawn about the estimation error from a finite number of samples.

The sampling rule for the maximization method under consideration and the convergence of the method are discussed in Chapter II. Chapter III describes the formulation of the computerized algorithm based on the procedures specified by the sampling rule and the convergence criteria. It will be described in Chapter IV how the algorithm, slightly modified, can be used to estimate the zeros of a function. Several sample problems, ranging in degree of difficulty, were selected and solved by the computerized algorithm resulting from this research in an attempt to test the method and as a basis for comparison with the solutions obtained by other methods. The sample problems and their solutions are discussed in Chapter V. An experiment was performed to test the algorithm's sensitivity to the Lipschitzian constant and the shape of the objective function. The experimental procedure, results and conclusions are discussed in Chapter VI. The results and conclusions of this research are discussed in Chapters VII and VIII and the recommendations for future research are discussed in Chapter IX. Appendices A - F contain the flow charts and program listings for the method under consideration.

II. DEVELOPMENT OF THE SAMPLE RULE FOR THE MAXIMIZATION METHOD TO BE CONSIDERED

Consider the sequence of sampling points $\{ x_0, x_1, \ldots, x_n \}$ in $[a, b]$ and their corresponding sample values $\{ f (x_0), f (x_1) \ldots, f (x_n) \}$. By (8)

$$f (x) \leq f (x_0) + C \mid x - x_0 \mid$$

$$f (x) \leq f (x_1) + C \mid x - x_1 \mid$$

$$f (x) \leq f (x_n) + C \mid x - x_n \mid .$$

Define

$$F_n (x) = \min_{k = 0, 1, \ldots, n} \{ f (x_k) + C \mid x - x_k \mid \} , \tag{9}$$

to be the piecewise linear function passing through the points $(x_0, f (x_0))$, $(x_1, f (x_1))$, $\ldots$, $(x_n, f (x_n))$ with the slope determined by the Lipschitzian constant C, defined by (8). Figure 2 illustrates the function F_n defined by (9).

From Fig. 2 it can be seen that for n samples the whole function f is upper-bounded by F_n with at most $(n + 2)$ peaks (including possible peaks at the end-points of the interval $[a, b]$.)

Define

$$Z_n = \max_{x \in [a, b]} F_n (x) . \tag{10}$$

Clearly, Z_n will be the ordinate of the highest peak of F_n. Let

$$\varphi = \max_{x \in [a, b]} f (x) .$$

be the global maximum of the function f and let

$$\hat{\varphi}_n = \max \{ f (x_0), \ldots, f (x_n) \} \tag{11}$$

be the estimate of φ after n iterations (samples) have been observed.

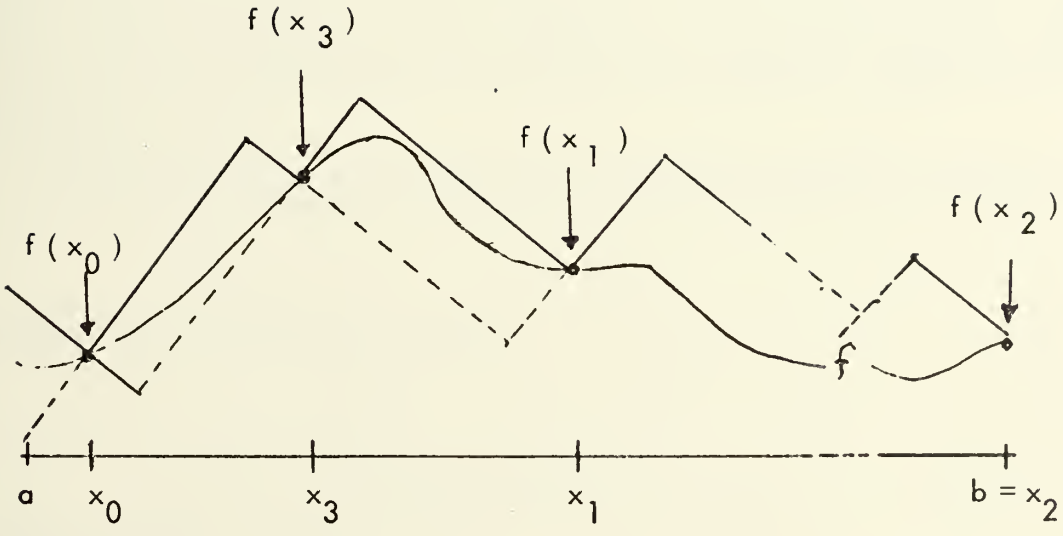


Figure 2. Graph of function

$$F_n(x) = \min_{k = 0, 1, \ldots, n} \{ f(x_k) + C \mid x - x_k \mid \} .$$

Since F_n upperbounds f and $\hat{\varphi}_n$ is the largest sample value observed so far, the following is concluded:

$$\hat{\varphi}_n \leq \varphi \leq Z_n . \tag{12}$$

It seems plausible at this point to utilize the difference between Z_n and $\hat{\varphi}_n$ as a basis for selecting the sample rule under consideration. Define

$$e_n = Z_n - \hat{\varphi}_n \tag{13}$$

as the maximum possible error between φ and $\hat{\varphi}_n$ after n observations. Furthermore, let ξ_n be the abscissa of Z_n. Since by (12) the global maximum φ is between Z_n and $\hat{\varphi}_n$, it follows that as e_n defined by (13) decreases, the global maximum φ can be more accurately determined.

The method under consideration seeks that choice of x_{n+1} that will minimize e_n. ξ_n is the optimal selection in the minimax sense, in that any other choice of x_{n+1} could fail to decrease e_n by the same or larger amount.

Since the above procedure is both sequential and optimal in a minimax sense, the sampling rule of selecting the abscissa of the highest peak in F_n is used for the method under consideration. That the sampling rule for the method under consideration is in fact optimal relative to the class of all functions that are Lipschitzian is proved by Shubert [Ref. 10].

The sampling sequence for this method is defined mathematically as follows:

$$x_0 \in [a, b]$$

where x_0 is the initial sampling point,

$$x_{n+1} \text{ such that}$$

$$F_n(x_{n+1}) = Z_n, \quad n = 0, 1, \ldots,$$

otherwise arbitrary, where

$$Z_n = \max_{x \in [a, b]} F_n(x),$$

$$F_n(x) = \min_{k = 0, 1, \ldots, n} \{ f(x_k) + C | x - x_k | \} .$$

The nature of the sampling sequence for this method suggests that as $n \to \infty$, $\hat{\varphi}_n \uparrow \varphi$, $Z_n \downarrow \varphi$, and $e_n \to 0$. Shubert [Ref. 10] theoretically proves that these conditions are satisfied for the sampling rule just considered.

The rate at which the error, e_n, defined by (13) approaches zero is worthy of consideration at this point. Shubert [Ref. 10] has shown that the slowest possible rate at which e_n approaches zero occurs when f = constant, for any arbitrary selection of the initial sampling point x_0. It remains to be seen how fast e_n approaches zero when the initial sampling point is

$$x_0 = \frac{(a+b)}{2}$$

The speed of convergence in light of the conditions specified above will now be studied.

Suppose

$$\Psi = \{ x \in [a, b] : f(x) = \varphi \} \tag{14}$$

is the set of all x for which the global maximum is attained. Furthermore, since the case is being considered for f = any constant, let the constant be zero for ease of illustration. Define $C > 0$ as the value of the Lipschitzian constant. By definition $f(x) = 0$ for every $x \in [a, b]$, hence, $\hat{\varphi}_n = 0$ for all n and

$$\Psi = [a, b].$$

A further implication about e_n defined by (13) can be made since

$$e_n = Z_n - \hat{\varphi}_n$$

and $$\hat{\varphi}_n = 0$$

for all n then $e_n = Z_n$ for all n.

Utilizing the function F_n defined by (9) and the sampling rule for the method under consideration, a general pattern for the rate at which the error e_n decreases can be determined. Figure 3 illustrates the sampling sequence when f = constant.

It can be seen from Fig. 3 that each sampling point after n = 2 increases the number of peaks in F_{n+1} by two and that the two new peaks created are equal in height. Furthermore, it can be easily shown that the height of these two new peaks is equal to half the height of the peak that created them, see Fig. 4. The calculations illustrated in Fig. 4 are as follows:

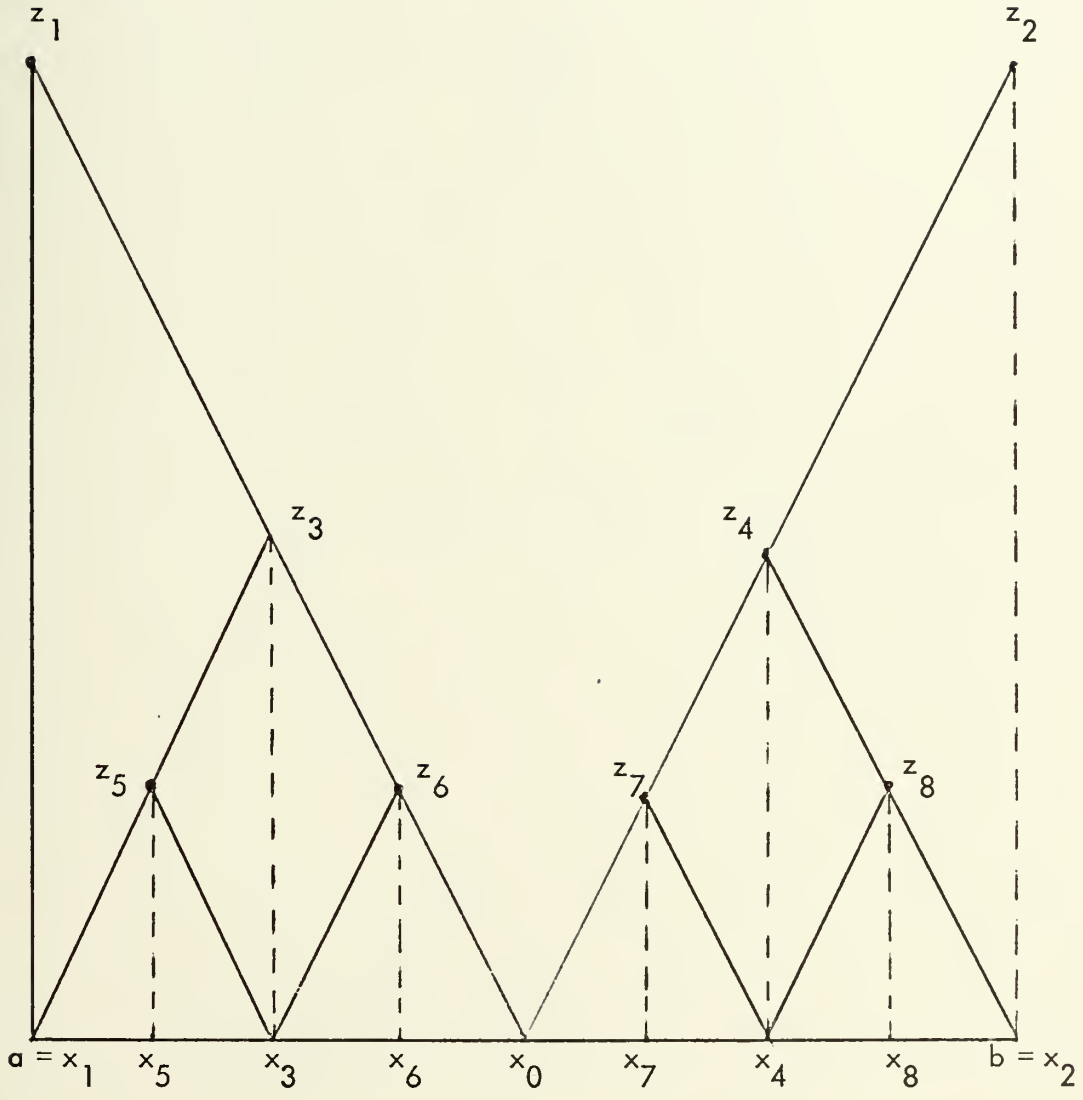


Figure 3. Illustration of sampling sequence when f = 0.

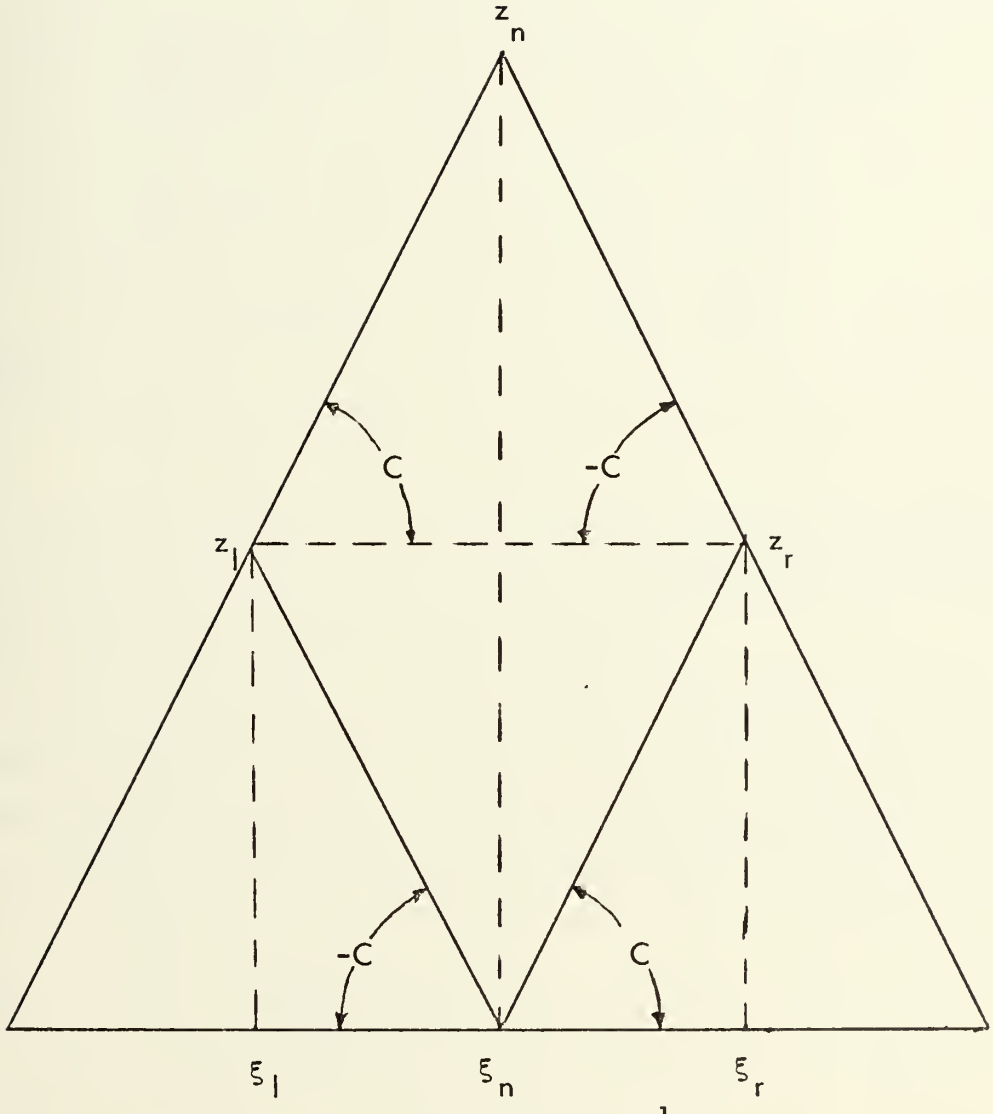


Figure 4. Illustration to show that $z_l = z_r = \frac{1}{2} z_n$

$$C = \frac{Z_n - Z_1}{\xi_n - \xi_1} \qquad C = \frac{Z_r}{\xi_r - \xi_n}$$

$$-C = \frac{Z_1}{\xi_1 - \xi_n} \qquad -C = \frac{Z_n - Z_r}{\xi_n - \xi_r}$$

$$\frac{Z_n - Z_1}{\xi_n - \xi_1} = \frac{Z_1}{\xi_n - \xi_1} \qquad \frac{Z_r}{\xi_r - \xi_n} = \frac{Z_n - Z_r}{\xi_r - \xi_n}$$

$$Z_n - Z_1 = Z_1 \qquad Z_r = Z_n - Z_r$$

$$Z_1 = \frac{Z_n}{2} \qquad Z_r = \frac{Z_n}{2}.$$

Hence,

$$Z_1 = Z_r = \frac{Z_n}{2}.$$

The results of the computations performed above are tabulated in Table I, for $n = 16$. The purpose of Table I is to outline the sampling sequence so that a general pattern of the way e_n decreases can be determined.

Let k be the length of the interval during which the value of Z_n does not change. Then after two samples have been taken, exclusive of the initial one, it can be seen from Table I that k increases with powers of two while Z_n decreases at the same rate.

The above reasoning can be stated mathematically in the following manner :

If $$2^k < n \leq 2^{k+1},$$

then $$Z_n = \frac{C\,(b - a)}{2^{k+1}}.$$

$$2^{k+1} \geq n$$

implies that

$$\frac{1}{2^{k+1}} \leq \frac{1}{n}.$$

	n	z_n
	1	C (b - a) / 2
	2	C (b - a) / 2
k = 1	3	C (b - a) / 4
k = 1	4	C (b - a) / 4
k = 2	5	C (b - a) /8
k = 2	6	C (b - a) / 8
k = 2	7	C (b - a) / 8
k = 2	8	C (b - a) / 8
k = 3	9	C (b - a) / 16
k = 3	10	C (b - a) / 16
k = 3	11	C (b - a) / 16
k = 3	12	C (b - a) / 16
k = 3	13	C (b - a) / 16
k = 3	14	C (b - a) / 16
k = 3	15	C (b - a) / 16
k = 3	16	C (b - a) / 16

Table 1. Illustration of general pattern for decreasing e_n.

Hence

$$Z_n \leq \frac{C(b-a)}{n},$$

and it follows that if f = constant, the error, e_n, decreases at least as fast as

$$\frac{C(b-a)}{n}.$$

Of course, the estimation error resulting from a nonsequential (random or grid) search would also decrease as $\frac{1}{n}$. However, as experimental results of Chapter V indicate, the actual rate of the sequential algorithm is typically much faster.

To locate the global maximum in the experimental region it is necessary to determine the set Ψ of all $x \in [a, b]$ at which the global maximum is attained.

Let

$$\Psi_n = \{ x \in [a, b] : F_n(x) \geq \hat{\varphi}_n \}, \tag{15}$$

for n = 0, 1, ..., where F_n is the function defined by (9). Since

$$\hat{\varphi}_n \leq \hat{\varphi}_{n+1} \leq \varphi$$

and $$f(x) \leq F_{n+1}(x) \leq F_n(x),$$

for every $x \in [a, b]$ it follows that

$$\Psi \subset \Psi_{n+1} \subset \Psi_n, \; n = 0, 1, \ldots . \tag{16}$$

Figure 5. illustrates the heuristic approach to locating the global maximum φ. Consider the situation after (n + 1) samples have been observed. The heavy solid lines in Fig. 5 represent the uncertainty in the location of the maximum if $\hat{\varphi}_{n+1} \neq \hat{\varphi}_n$. If $\hat{\varphi}_{n+1} = \hat{\varphi}_n$ then the intervals of uncertainty remain the same and only $F_n(x)$ changes. However, if $\hat{\varphi}_{n+1} = f(x_{n+1})$, then the estimate moves closer to φ from below and as a result the intervals of uncertainty will be reduced in length and more accurately determine the abscissae for locating φ.

Clearly, Ψ is the smallest subset of [a, b] that defines the location of φ, Ψ_{n+1} contains all elements of Ψ but may contain several elements that do not lead to the location of φ. Finally, the largest uncertainty set obtained from the sampling sequence is Ψ_n. Thus,

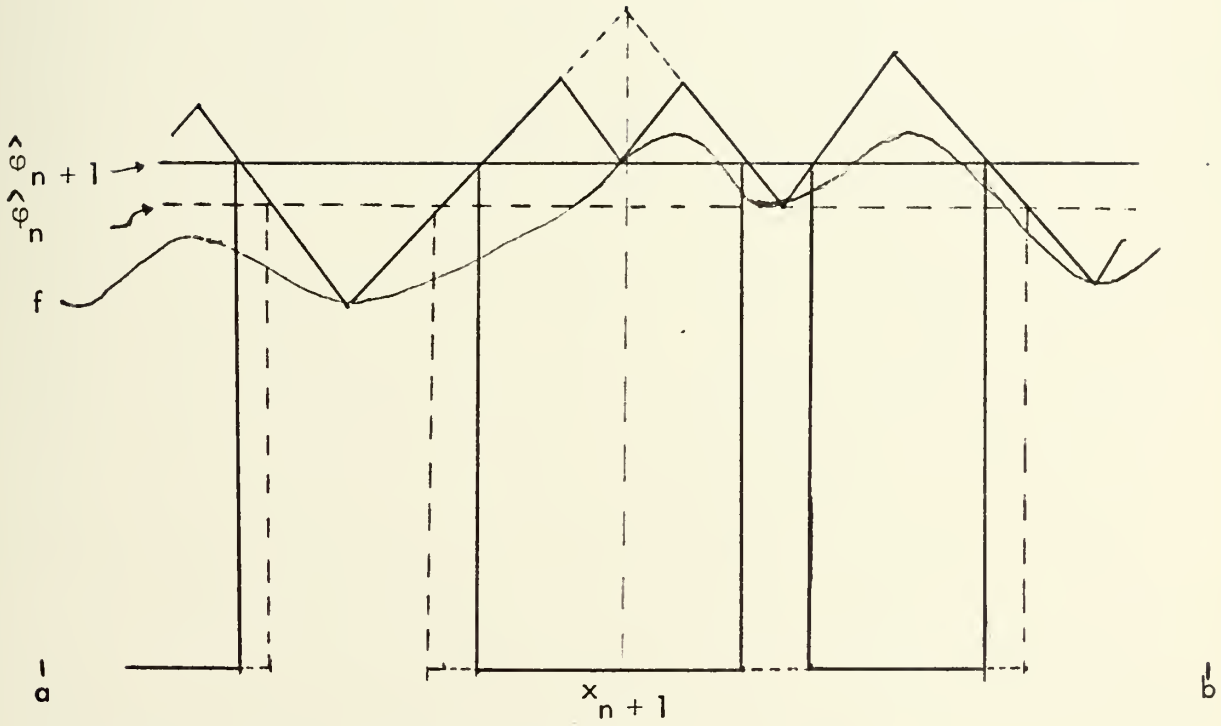


Figure 5. Locating the global maximum.

$\Psi \subset \Psi_{n+1} \subset \Psi_n$, $n = 0, 1, \ldots$, defined by (15). Furthermore, it has been shown by Shubert [Ref. 10] that the Lebesque measure of $\Psi_n - \Psi$ converges to zero as $n \to \infty$.

III. DEVELOPMENT OF THE COMPUTERIZED ALGORITHM

A. PRELIMINARY CONSIDERATIONS

The sampling rule developed in Chapter II lends itself nicely to the formulation of a computerized algorithm for estimating the global maximum and its location of any deterministic function of a single real variable defined in a closed interval [a, b].

The information gathered from all samples obtained so far must be stored in the computer's memory so that it can make the proper decision as to where the next sampling point should be located.

Define the data set stored in the computer's memory after n samples have been observed as

$$D_n = \{ (t_1, z_1), (t_2, z_2), \ldots, (t_{H_n}, z_{H_n}); \hat{\varphi}_n \}$$

where $z_1 \leq z_2 \leq \ldots \leq z_{H_n}$, and the vectors (t_i, z_i), $i = 1, 2, \ldots, H_n$, are the coordinates of the maxima of F_n defined by (9).

Let

$$x_0 = \frac{1}{2}(a + b)$$

be the initial sampling point and $f(x_0)$ the corresponding sample value. Furthermore,

$$Z_a = f(x_0) - C\,|a - x_0|,$$

$$Z_b = f(x_0) + C\,|b - x_0|.$$

Clearly, z_a and z_b are the two maxima of $F_0(x)$ defined by (9) at the endpoints of the experimental region [a, b]. $z_a = z_b$ since $(b - x_0) = -(a - x_0)$ implies that $z_b = f(x_0) - C(a - x_0) = z_a$, see Fig. 6. $\hat{\varphi}_0 = f(x_0)$ is defined as the estimate of φ after the zero (th) iteration. Let $t_a = a$, and $t_b = b$ be the abscissae of z_a and z_b respectively.

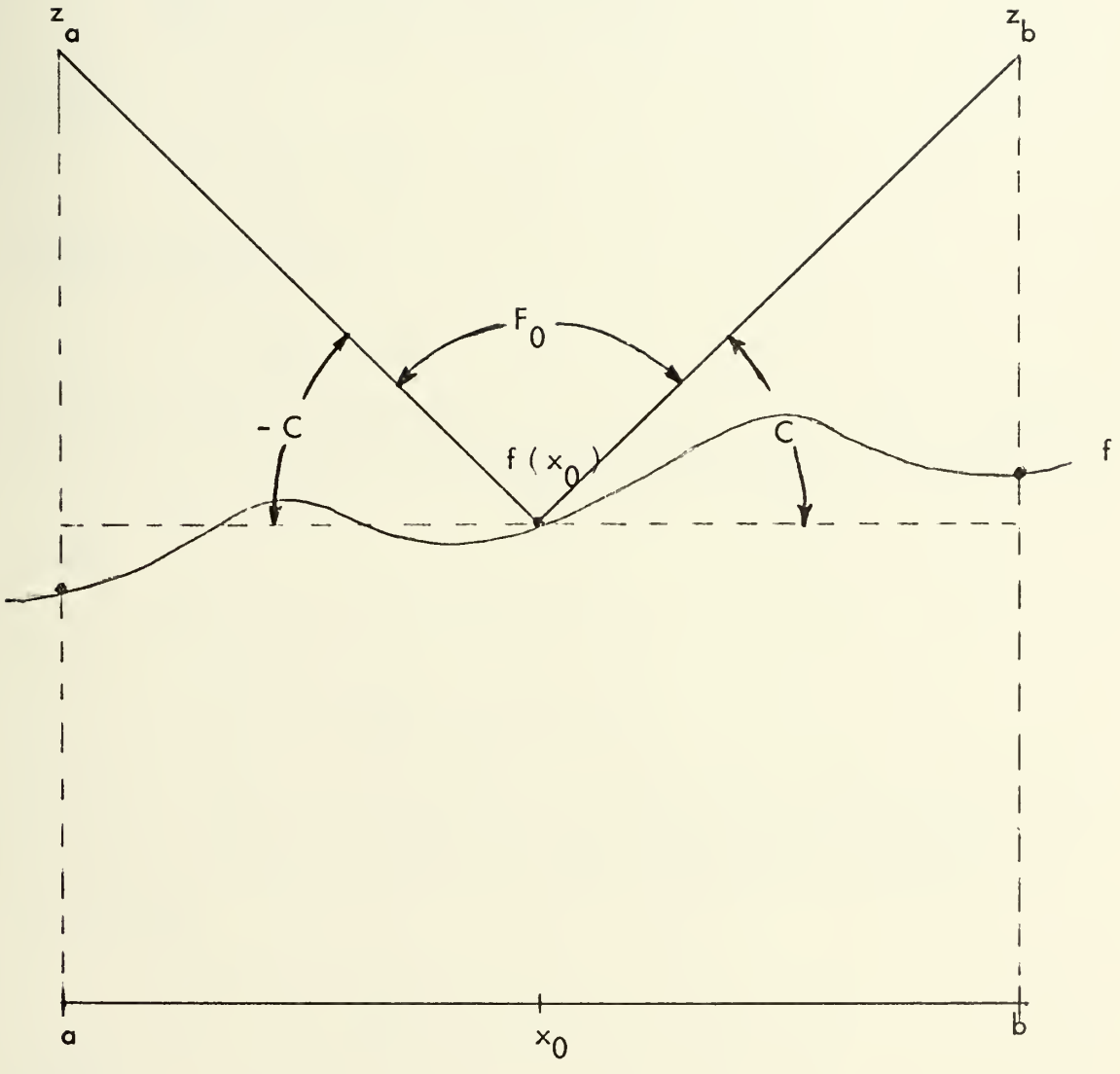


Figure 6. Zeroth iteration.

B. FIRST ITERATION

Define $F_0(x)$ as the piecewise linear function connecting the points (a, z_a), $(x_0, f(x_0))$, (b, z_b). For this iteration, maxima of $F_0(x)$ consist of the vectors (a, z_a), (b, z_b). Since $z_a = z_b$, the arrangement of the vectors in D_0 is completely arbitrary. Suppose $t_{H_0} = t_a$, then

$$D_0 = \{ (b, z_b), (a, z_a); \hat{\varphi}_0 \} .$$

By (10)

$$z_0 = z_{H_0} = z_a$$

and by the sampling rule

$$x_1 = t_{H_0} = a.$$

The corresponding sample value for $x_1 = a$ is $f(x_1) = f(a)$. Drop the vector $(t_{H_0}, z_{H_0}) = (a, z_a)$ from the data set D_0 and add the new vector (t_r, z_r) where

$$z_r = \frac{1}{2}(z_a + f(a))$$

$$t_r = a + \frac{1}{2C}(z_a - f(a)),$$

see Fig. 6. The new set of data D_1 is then obtained by rearranging the vectors (b, z_b) and (t_r, z_r) in the order of nondecreasing second component. It is obvious that

$$z_r \leq z_a = z_b .$$

Thus,

$$D_1 = \{ t_r, z_r), (b, z_b); \hat{\varphi}_1 \}$$

where $\hat{\varphi}_1 = \max \{ \hat{\varphi}_0, f(a) \}$. D_1 is again the set of coordinates for all maxima of $F_1(x)$ defined by the piecewise linear function connecting the points $(a, f(a))$, (t_r, z_r), $(x_0, f(x_0))$, $(b, f(b))$, see Fig. 7.

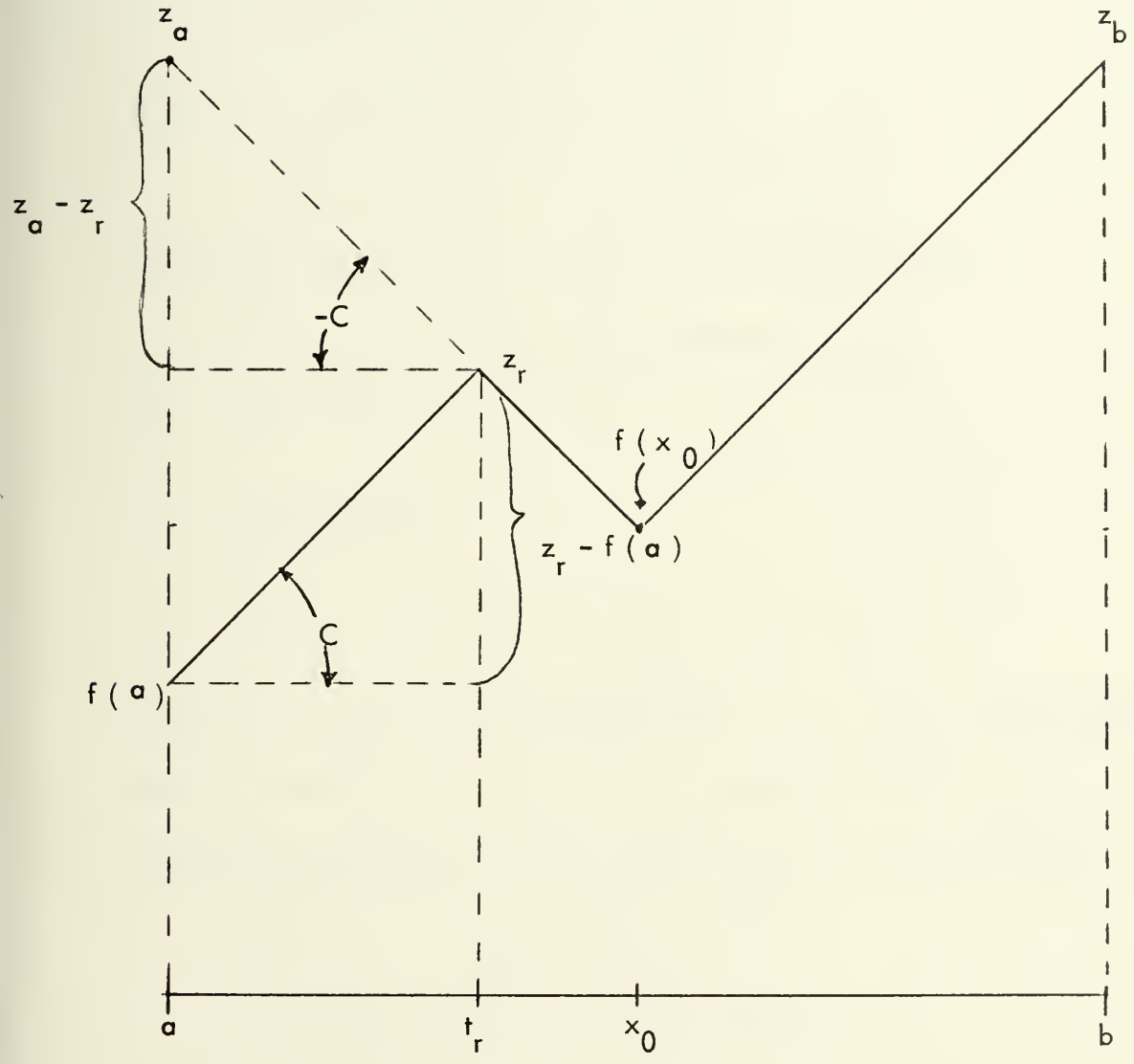


Figure 7. First iteration.

C. SECOND ITERATION

By (10) $z_1 = z_{H_1} = z_b$ and by the sampling rule $x_2 = t_{H_1} = b$. The corresponding sample value for $x_2 = b$ is $f(b)$. Drop the vector $(t_{H_1}, z_{H_1}) = (b, z_b)$ from the data set D_1 and add the new vector (t_l, z_l) where

$$z_l = [\, z_b + f(b) \,] / 2$$
$$t_l = b - [\, z_b - f(b) \,] / 2C$$

see Fig. 8. Arrange the data set such that

$$D_2 = \{ (t_l, z_l), (t_{H_2}, z_{H_2}); \hat{\varphi}_2 \}$$

where $z_{H_2} = \max \{ z_l, z_r \}$, t_{H_2} the corresponding abscissa;

$z_l = \min \{ z_l, z_r \}$, t_l the corresponding abscissa; and

$\hat{\varphi}_2 = \max \{ \hat{\varphi}_1, f(t_{H_2}) \}$.

D. n (th) ITERATION

For $n \geq 3$,

$$D_n = \{ (t_1, z_1), \ldots, (t_{H_n}, z_{H_n}); \hat{\varphi}_n \}.$$

By (10), $z_n = z_{H_n}$ and by the sampling rule $x_{n+1} = t_{H_n}$. The corresponding sample value for $x_{n+1} = t_{H_n}$ is $f(t_{H_n})$. Drop the vector (t_{H_n}, z_{H_n}) from the data set and add the two new vectors (t_l, z_l), (t_r, z_r) where

$$z_l = z_r = [\, z_{H_n} + f(t_{H_n}) \,] / 2$$

$$t_l = t_{H_n} - [\, z_{H_n} - f(t_{H_n}) \,] / 2C$$

$$t_r = t_{H_n} + [\, z_{H_n} - f(t_{H_n}) \,] / 2C$$

see Fig. 9. The new set of data, D_{n+1}, is obtained by rearranging the vectors $(t_1, z_1), \ldots, (t_{H_n - 1}, z_{H_n - 1}), (t_l, z_l), (t_r, z_r)$ in the order of

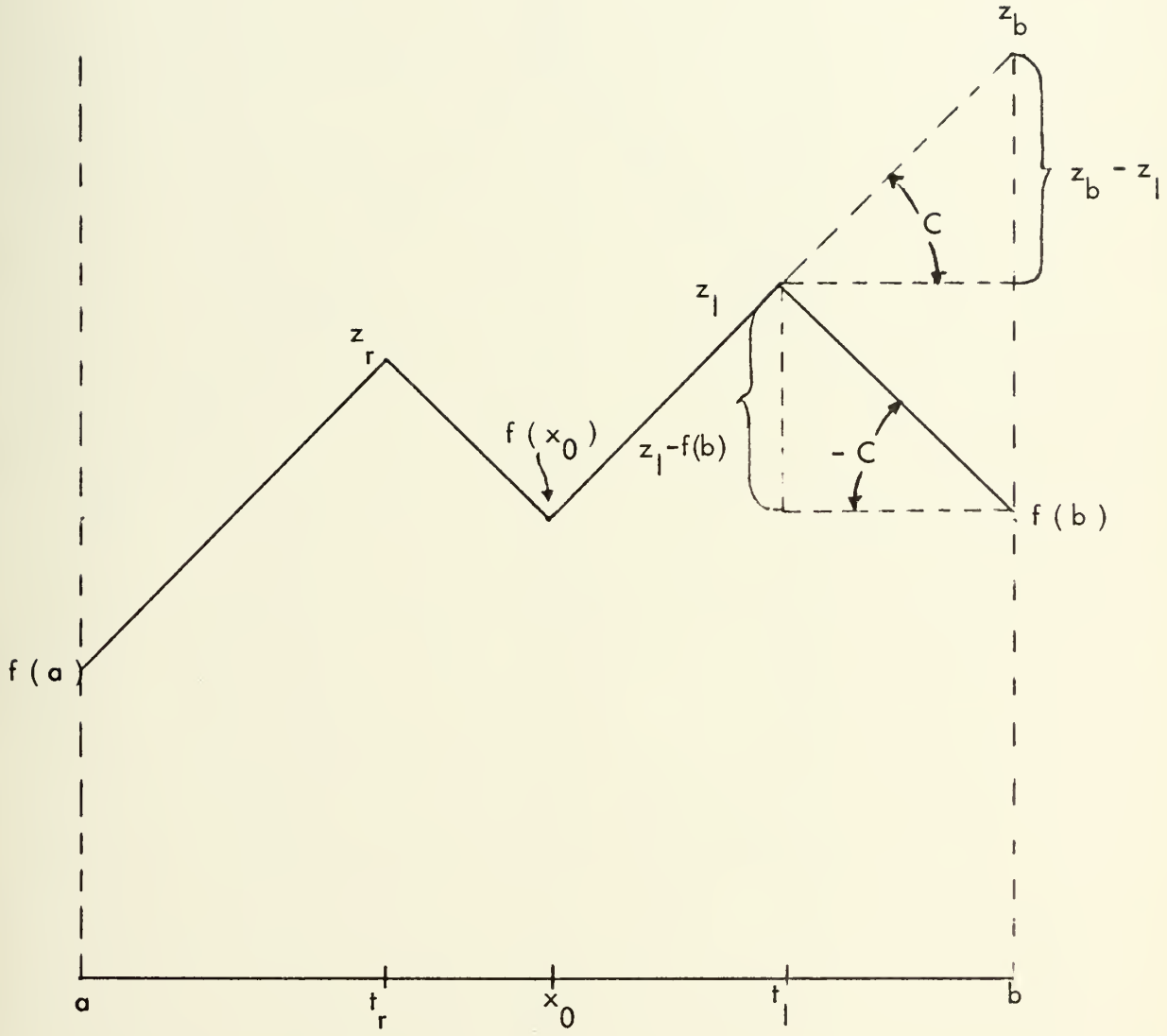


Figure 8. Second iteration.

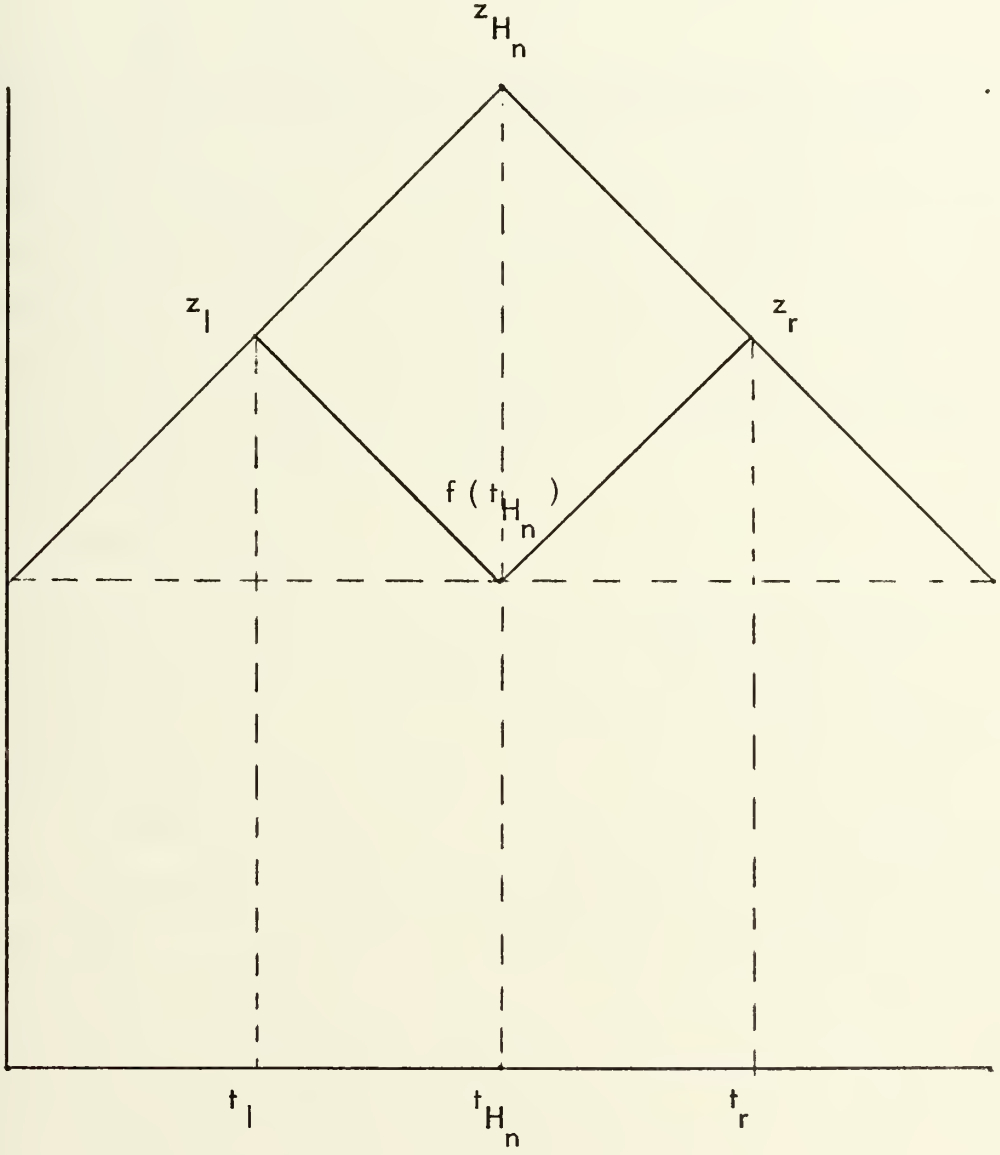


Figure 9. nth iteration.

nondecreasing second component. Thus, D_{n+1} is the set of coordinates for all the maxima of F_{n+1} defined by (9) and $\hat{\varphi}_{n+1} = \max \{ \hat{\varphi}_n, f(t_{H_n}) \}$.

E. STOPPING RULE

The iterations continue until

$$z_{H_n} - \hat{\varphi}_n \leq \varepsilon ,$$

where $\varepsilon > 0$ is the desired accuracy in estimating the global maximum φ, and $\hat{\varphi}_n$ is the largest value for f observed so far, see Fig. 10. The experimenter should be realistic when he specifies ε. Naturally, he would like the error to be zero, however, this would be feasible in a finite number of iterations only if the function f coincides with the function F_n in the area of the maximum. Since this is usually not the case, he must be willing to accept some error between φ and $\hat{\varphi}_n$ that will allow the computer to execute the algorithm in a reasonable amount of time. Cost will increase as more accuracy is desired.

F. REDUCTION OF DATA

The main computational disadvantage of the algorithm as described so far is that from the second iteration on the memory content increases by one vector (t_i, z_i) at each iteration, i .e., $H_n = n$ for $n \geq 3$. This can be remedied to some extent by dropping at each iteration all vectors $(t_i, z_i) \in D_n$ such that $z_i < \hat{\varphi}_{n+1}$, see Fig. 11. Although the function F_n is no longer being stored completely, it is easy to see that the sequence of sampling points generated by the algorithm remains the same as before. It is clear from Fig. 11 that since F_n upperbounds f and $\varphi \geq \hat{\varphi}_{n+1}$, the abscissae of all maxima of F_n below $\hat{\varphi}_{n+1}$ will never be sampled anyway, due to the very nature of the sampling procedure. The shape of the function in the experimental region will determine how fast the memory content increases. It has been shown in Chapter II that in the most unfavorable case, f = constant, it will increase by one vector per iteration.

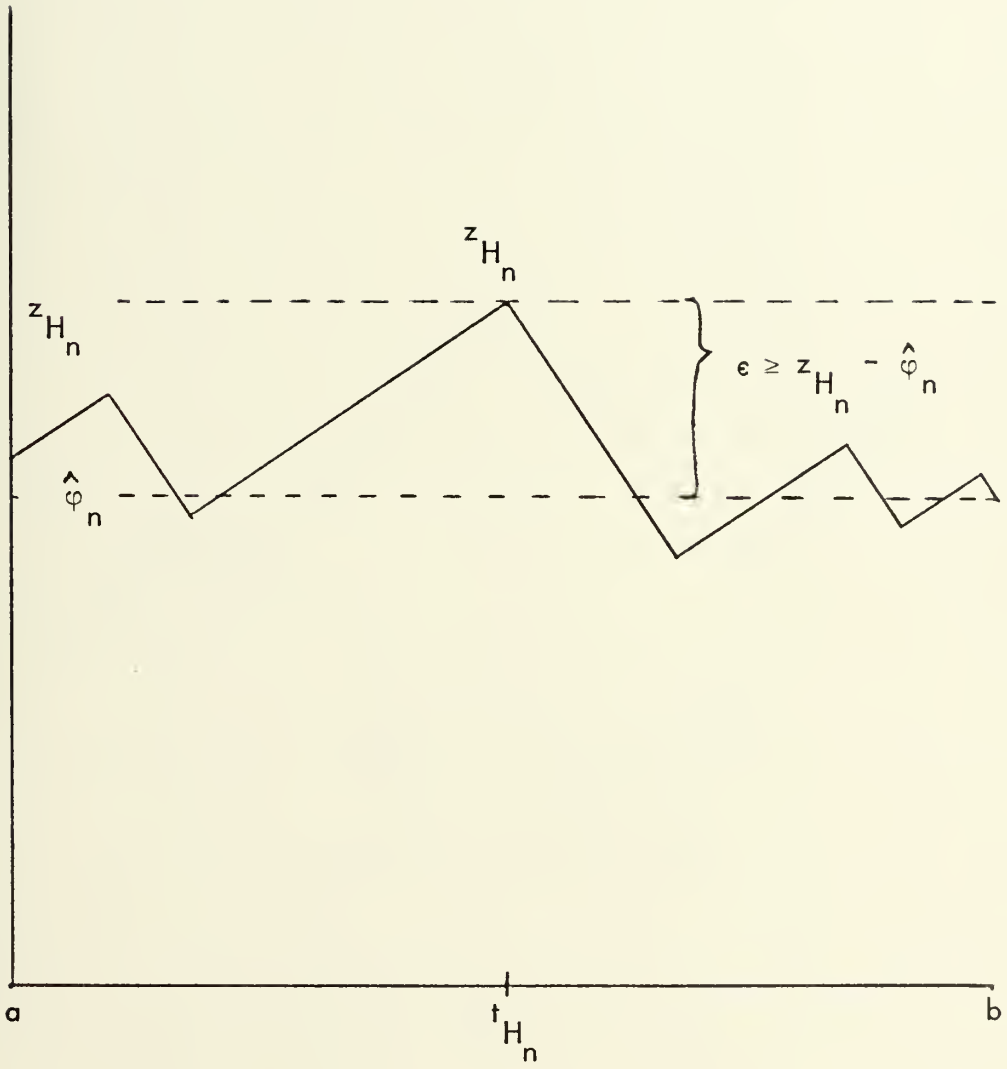


Figure 10. Stopping Rule

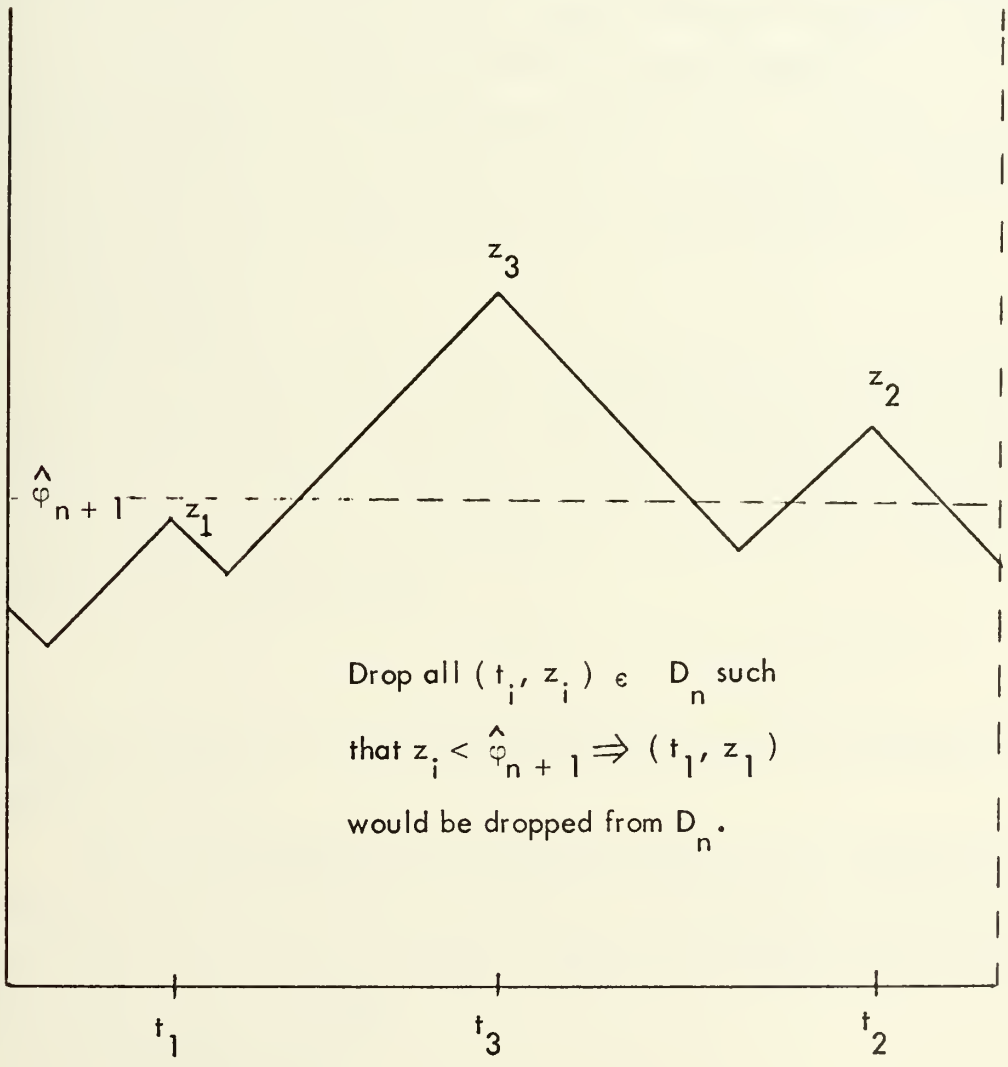


Figure 11. Reduction of data.

G. LOCATING THE GLOBAL MAXIMUM

Define D_ε as the data set

$$D_{n+1} = \{ (t_1, z_1), \ldots, (t_{H_n+1}, z_{H_n+1}); \hat{\varphi}_{n+1} \}$$

in the computer's memory after the final iteration of the algorithm as described so far. Furthermore, define F_ε as the corresponding function of F_{n+1} defined by (9). D_ε then contains coordinates for all maxima of F_ε which upperbounds the function $f(x)$, i.e., $f(x) \leq F_\varepsilon(x)$ by (9). Thus, the second components z_i of all $(t_i, z_i) \in D_\varepsilon$ are ε - close to φ and are at least as great as $\hat{\varphi}_\varepsilon = \hat{\varphi}_{n+1}$ after the last iteration. The information contained in D_ε provides many alternatives for estimating the location (s) of φ in [a, b], depending on the desires of the experimenter. Four of these alternatives will be examined below.

1. Alternative I

This alternative could be used if the experimenter is interested in a single estimate for the location of φ in [a, b]. Only a slight modification to the established algorithm is necessary to obtain these results. Simply carry along the abscissa of the largest sample value obtained so far in the data set D_n. Thus, the revised data set at each iteration becomes

$$D_n = \{ (t_1, z_1), \ldots, (t_{H_n}, z_{H_n}); (x_{\hat{\varphi}_n}, \hat{\varphi}_n) \} .$$

At the final iteration $(\hat{x}_\varepsilon, \hat{\varphi}_\varepsilon)$ will be the coordinates of the estimated φ and its location in [a, b].

The major drawback to this alternative is that it fails to estimate the location of all global maxima in [a, b], however, it is relatively simple and particularly useful if the function is unimodal or when only a single estimate for the location of the maximum is desired.

2. ALTERNATIVE II

This alternative may be successful in estimating all locations of the global maximum in [a, b]. D_ε contains all information necessary to obtain these results.

From the set of first components of (t_i, z_i) in D_ε, obtain a new data set

$$G = \{ (t_1, y_1), \ldots, (t_{H_\epsilon}, y_{H_\epsilon}); \hat{\varphi}_\epsilon \},$$

where $y_i = f(t_i)$, $i = 1, \ldots, H_\epsilon$.

Define

$$T = \{ t_i : (t_i, y_i) \in G, y_i \geq \hat{\varphi}_\epsilon \}$$

as the set of points estimating the location of φ in $[a, b]$, see Fig. 12.

The major drawback of this alternative is that an uncertainty set containing the locations of global maxima in $[a, b]$ cannot be precisely determined, $x \in T$ implies that

$$\varphi - f(x) \leq \epsilon,$$

however, the true locations may still be outside of T. Figure 12 illustrates why this method may not be successful in estimating the locations of all global maxima. For this particular example, $(t_3, y_3) \notin T$ because $y_3 < \hat{\varphi}_\epsilon$. Even though t_3 is close to an abscissa for which φ is attained it is not considered in this case. The abscissa t_2 is less close to an abscissa for which φ is attained than is t_2 and yet this value is considered an estimate by this method.

Despite the fact that the method fails in this respect, it may still be used, with some reservations, if more than one location estimate is desired. The program listing for this alternative is located in Appendix E.

3. Alternative III

This alternative results in the genuine uncertainty set of the smallest possible size. From the considerations at the end of Chapter II, it follows that this set is given by

$$V = \{ x \in [a, b] : F_\epsilon(x) \geq \hat{\varphi}_\epsilon \}.$$

From the piecewise linear character of the function F_ϵ it follows further that the set V is a union of a finite number of disjoint closed intervals

$$V = \bigcup_i [x_{l_i}, x_{r_i}].$$

The following computations are needed to obtain V from the algorithm: Rearrange all

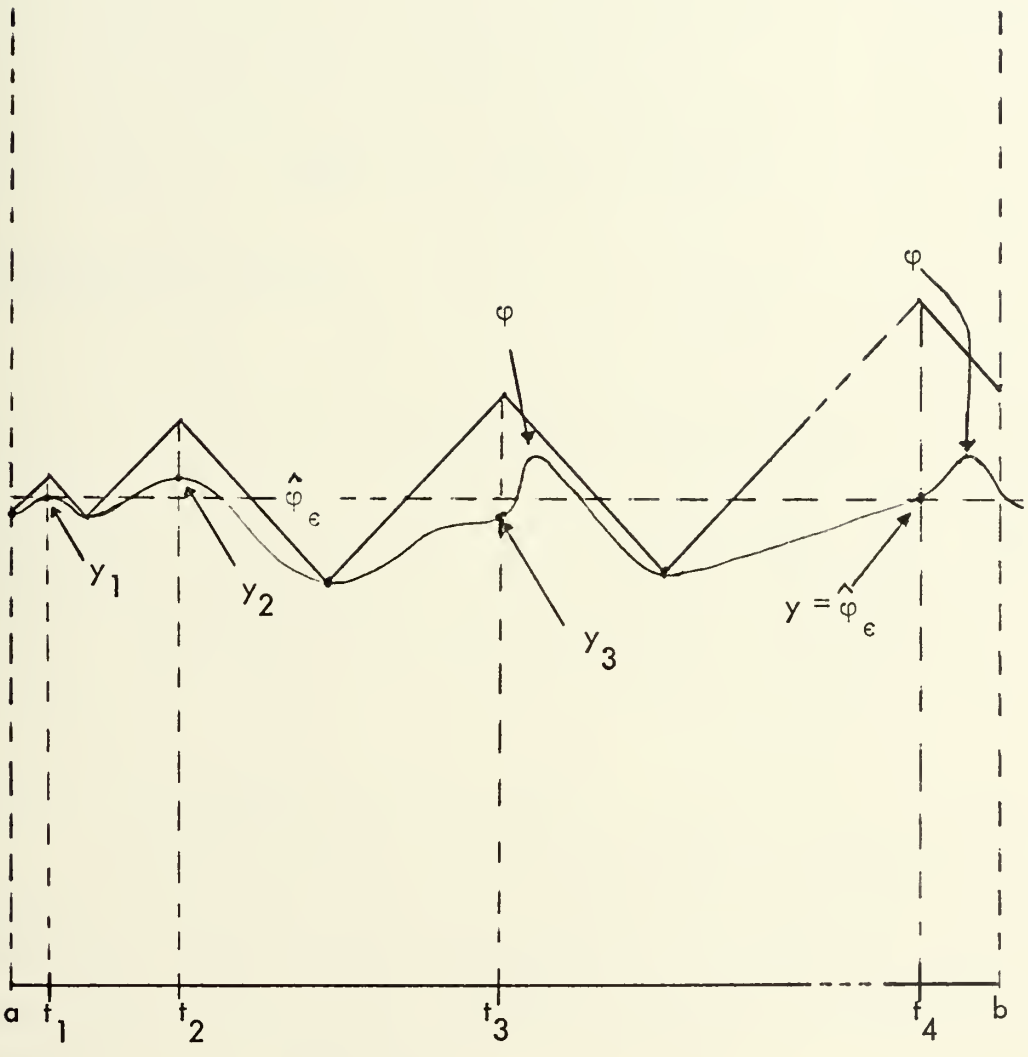


Figure 12. Alternative II.

$(t_i, z_i) \in D_\epsilon$ in nondecreasing order of first components. Then compute:

$$x_{l_1} = t_1 - \frac{1}{C}(z_1 - \hat{\varphi}_\epsilon)$$

unless x_{l_1} as computed above is less than the left most endpoint of the interval $[a, b]$ at which time x_{l_1} becomes a. That is,

$$x_{l_1} = \max \{ a, t_1 - \frac{1}{C}(z_1 - \hat{\varphi}_\epsilon) \}.$$

$$x_{r_i} = \min \{ b, t_i + \frac{1}{C}(z_i - \hat{\varphi}_\epsilon) \},$$

$$x_{l_{i+1}} = \max \{ a, t_i - \frac{1}{C}(z_i - \hat{\varphi}_\epsilon) \}.$$

The iterations continue until

$$x_{r_i} < x_{l_{i+1}}$$

at which time x_{r_i} becomes the endpoint of the first interval and $x_{l_{i+1}}$ the leftmost endpoint of the second interval. This procedure continues until $i = H_\epsilon$ at which time the rightmost endpoint of the last interval is computed to be $x_{r_i} = x_{r_{H_\epsilon}}$. The union of all these intervals belong to V. Clearly, $\Psi \subset V$. See Fig. 13 for an illustration of this approach.

Although this alternative provides the genuine uncertainty set of locations for φ, the number of intervals in the set becomes unwieldy and too clumsy to be practical. However, one can be assured that φ is contained in at least one of these intervals. The flow chart and program listing for this alternative are located in Appendices C and F respectively.

4. Alternative IV

This alternative is an attempt to reduce the number of intervals in V, for clarity, at the expense of enlarging the uncertainty set containing φ. The nature of the sampling rule is such that when the algorithm stops, the first components of $(t_i, z_i) \in D_\epsilon$ cluster about Ψ defined by (14). By rearranging the vectors

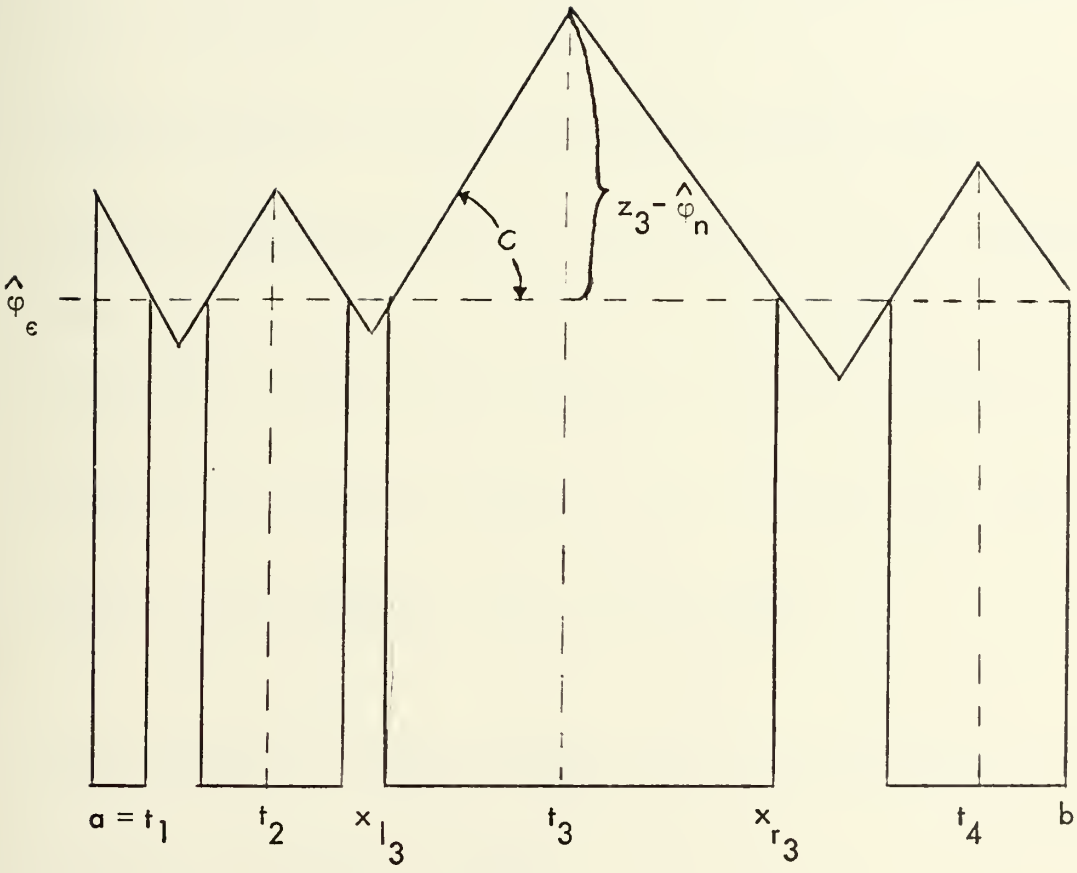


Figure 13. Sketch of Alternative III.

$(t_i, z_i) \in D_\epsilon$ in nondecreasing order of first component, the breakpoints between clusters are usually well defined as relatively large gaps between some t_i, t_{i+1} in the rearranged data set. Utilizing the endpoints of these clusters and the function F_ϵ, it is a very simple task to consolidate the points in each cluster to a single interval of uncertainty. The set of all such intervals formed in this manner is the set of intervals under consideration. Clearly, V is a subset of the union of these intervals, since the new set will include those portions of [a, b] between the intervals in V which do not contain φ. Hence, the new uncertainty set is larger than V but is more practical for presenting experimental results.

Let k be the number of clusters in the rearranged data set and let

$$H = \{ [u_{l_1}, u_{r_1}], \ldots, [u_{l_k}, u_{r_k}] \}$$

be the set of intervals such that u_{l_i} is the leftmost endpoint in the i th cluster and u_{r_i} is the rightmost endpoint of the i th cluster in H, i = 1, ..., k ; see Fig. 14. Clearly, $[u_{l_i}, u_{r_i}] \in H$ does not define the entire interval of uncertainty under consideration for the i th cluster. There is a portion of [a, b] to the left of u_{l_i} and a portion of [a, b] to the right of u_{r_i} which must be included in the uncertainty set under consideration. This is obvious because

$$\hat{\varphi}_\epsilon \leq \varphi \leq F_\epsilon$$

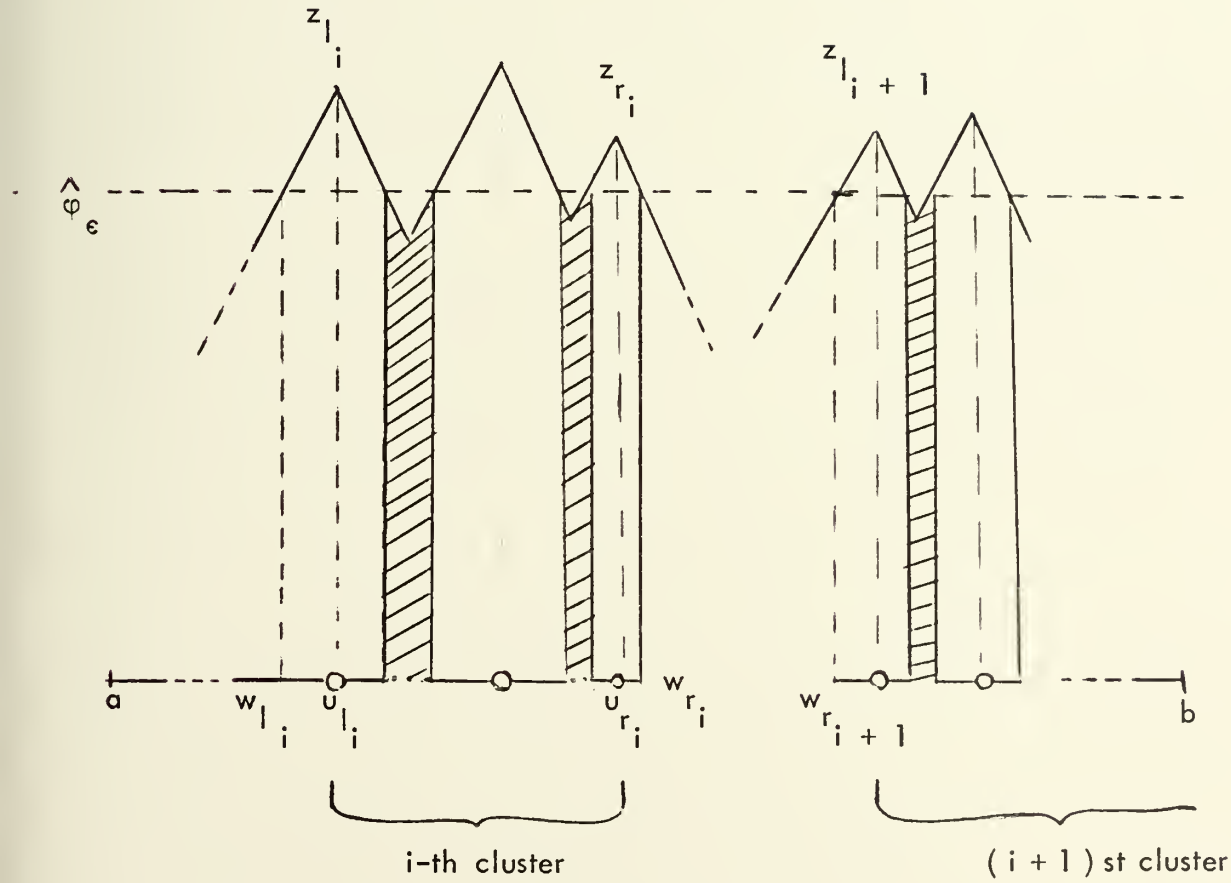
which implies that φ could very well be contained in the intervals $[w_{l_i}, u_{l_i}]$ and $[u_{r_i}, w_{r_i}]$, see Fig. 14, where

$$w_{l_i} = u_{l_i} - \frac{1}{C} (z_{l_i} - \hat{\varphi}_\epsilon)$$

$$w_{r_i} = u_{r_i} + \frac{1}{C} (z_{r_i} - \hat{\varphi}_\epsilon) .$$

The set of uncertainty intervals defined by

Figure 14. Sketch of uncertainty set W.



intervals forming the set V (Alt. III.)

added to V to obtain W (Alt. IV.)

o points forming the set T (Alt. II.).

$$W = \{ [w_{l_1}, w_{r_1}], \ldots, [w_{l_k}, w_{r_k}] \}$$

is the set desired for this alternative. The flow chart and program listing for this alternative have purposely been deleted from the Appendices since the intervals can be rather easily obtained from results of Alternative III.

IV. THE COMPUTERIZED ALGORITHM AS A MEANS FOR LOCATING ZEROS OF A FUNCTION

The algorithm resulting from this research can also be used to locate the zeros of a function. Only a slight modification to the algorithm is necessary to obtain these results.

By setting

$$g(x) = -|f(x)|,$$

the transformation is such that all global maxima φ of g in [a, b] occur at the zeros of f (x) . This fact allows the computerized algorithm to conform nicely to the task at hand. If the function f has at least one zero, the global maximum of g is $\varphi = 0$, therefore the new data set after each iteration will be

$$D_{n+1} = \{ (t_1, z_1), \ldots, (t_{H_n + 1}, z_{H_n + 1}); 0 \} .$$

The iterations continue as before until

$$z_{H_n} - \varphi \leq \epsilon \quad \text{or} \quad z_{H_n} \leq \epsilon$$

since $\varphi = 0$, at which time the iterations stop. The zeros of f (x) are estimated by applying the procedures discussed in Alternatives I - IV of Chapter III. The alternative used will depend on the form and the accuracy desired by the experimenter.

V. SAMPLE PROBLEMS

The results of the following sample problems were used as criteria for (1) testing the algorithm; and (2) comparison with results of the same problems obtained by other search procedures.

A. SAMPLE PROBLEM I

Given the deterministic function

$$f(x) = 3 + x - x^2$$

estimate the global maximum φ and its location in the experimental region [0, 2] with a desired accuracy of $\varepsilon = .01$.

1. Discussion

This problem was used throughout the design stages of the computerized algorithm due to its relative simplicity. It was primarily used in the design stages because the global maximum and its location could easily be determined by differential calculus.

Clearly, the function is unimodal in [0, 2], see Fig. 15, and by the calculus,

$$\varphi = 3.25,$$
$$\Psi = \{ .5 \} .$$

Since the function is differentiable, the Lipschitzian constant C can be computed as

$$C = \max_{x \in [a, b]} \left\{ \left| \frac{df}{dx} \right| \right\} = \max_{x \in [0, 2]} \{ | -2x + 1 | \} = 3 .$$

2. Results

For the desired accuracy of $\varepsilon = .01$, the algorithm required 63 iterations (samples) giving the estimate $\hat{\varphi}_\varepsilon = 3.25$. It was determined a priori from Fig. 15 that the number of global maxima was 1, hence, Alternative I was used to estimate the location of φ. The algorithm gave $\hat{x} = .5$ for this estimate. The largest number of vectors, (t_i, z_i), stored in the computer's memory was 40 and the computer time necessary (including the time required to compute $f(x_n)$) was 4.36 seconds.

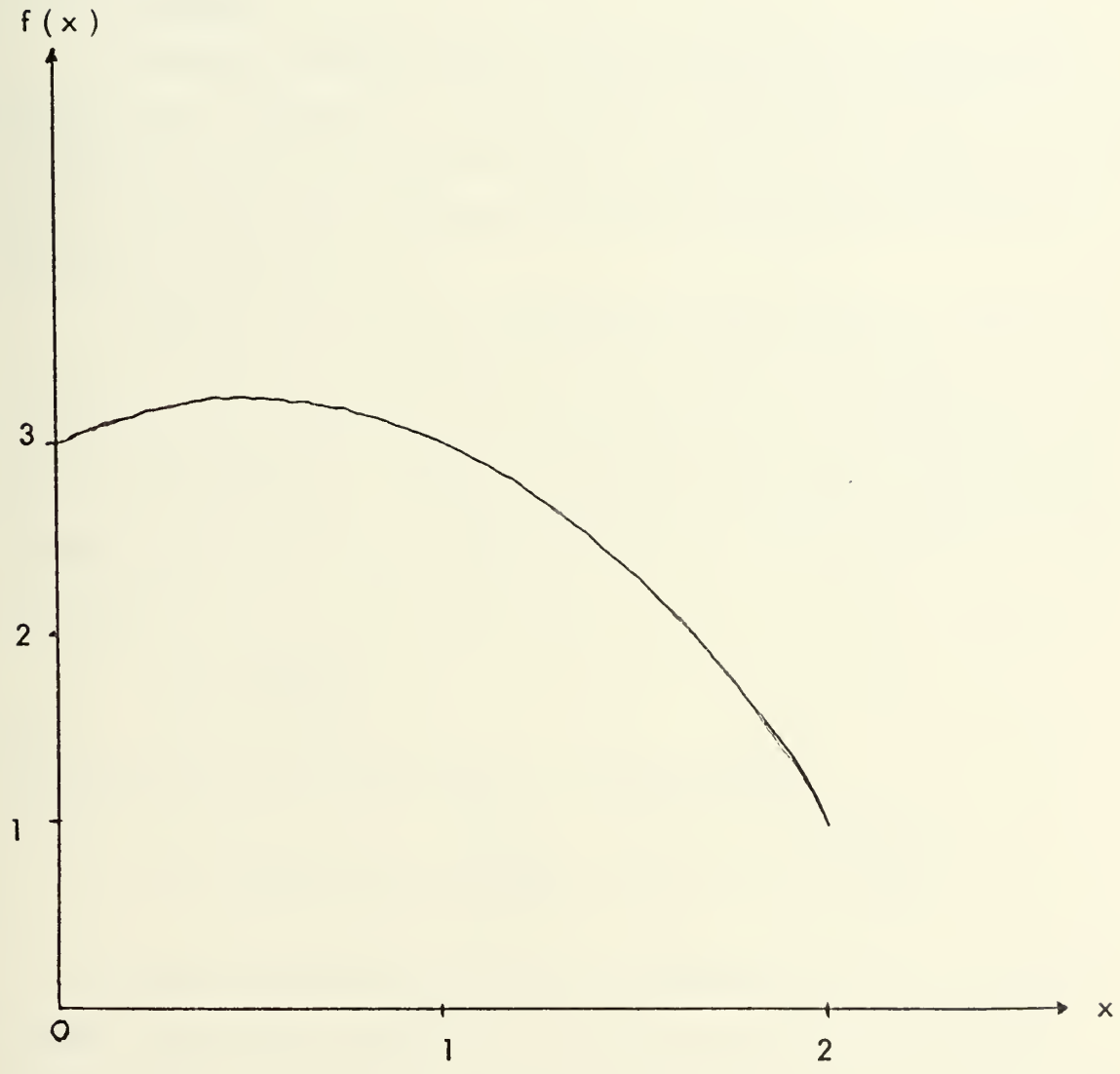


Figure 15. Graph of $f(x) = 3 + x - x^2$.

B. SAMPLE PROBLEM II

Given the function

$$f(x) = \sum_{i=1}^{5} \sin[-(i+1)\sqrt{x} + i],$$

estimate the global maximum φ and its location in [.01, 10] with a desired accuracy of ε = .01.

1. Discussion

The results of Sample Problem I seem to indicate that the computerized algorithm under consideration is quite successful in maximizing a unimodal function. It remains to be seen how the algorithm reacts to functions which are multimodal in nature. The problem described above was used to test the algorithm's ability to maximize multimodal functions.

The plotting package of an IBM 360/57 computer was used to plot f in [.01, 10], see Fig. 16. It is obvious from Fig. 16 that f is multimodal in [.01, 10] and that the global maximum occurs in the interval [.01, 1]. Although the function f is differentiable, it is difficult to obtain the exact global maximum and its location using the techniques of differential calculus. Hence, gradient techniques were applied in [.01, 1.0] to obtain

$$\varphi = \underline{12.0313}\ldots.$$

at $$\Psi = \{\ 0.2416\ \ldots\ \}\ .$$

The fact that f is differentiable allows C to be computed as

$$C = \max_{x \in [\ .01,\ 10\ ]} \left\{ \left| \sum_{i=1}^{5} -(i+1)\, x^{-1/2} \right| \right\} = 350.$$

For sake of illustration, the sequence of sampling points x_n vs. the iteration number n were plotted, see Fig. 17. It shows how the search soon abandons regions of [a, b] where f (x) is low and concentrates on search in the vicinity of maxima. The number of vectors stored vs. the number of iterations required were also plotted to illustrate the effects of the data reduction technique utilized by the proposed algorithm, see Fig. 18.

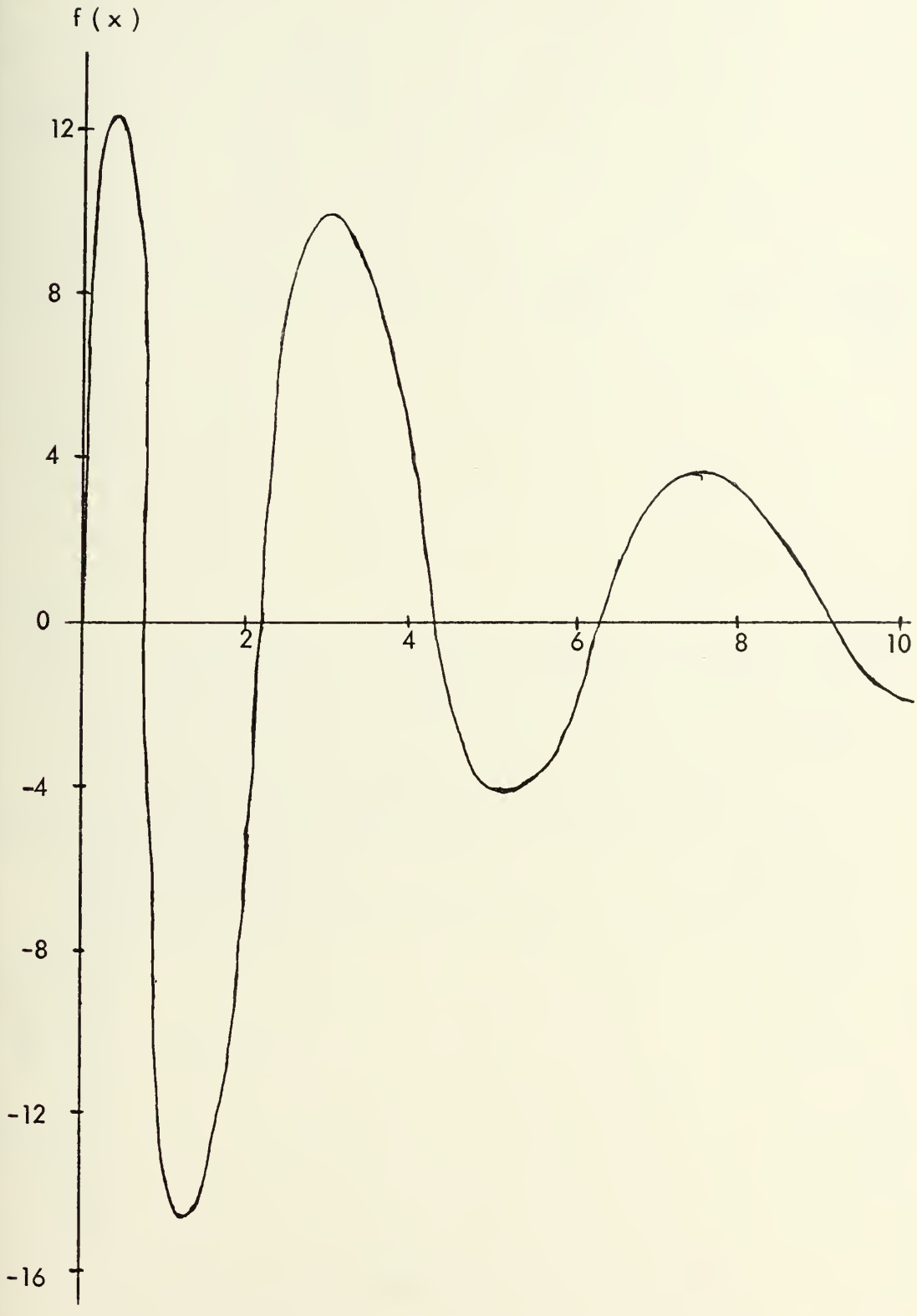


Figure 16. Graph of

$$f(x) = \sum_{i=1}^{5} i \sin\left[-(i+1)\sqrt{x} + i \right].$$

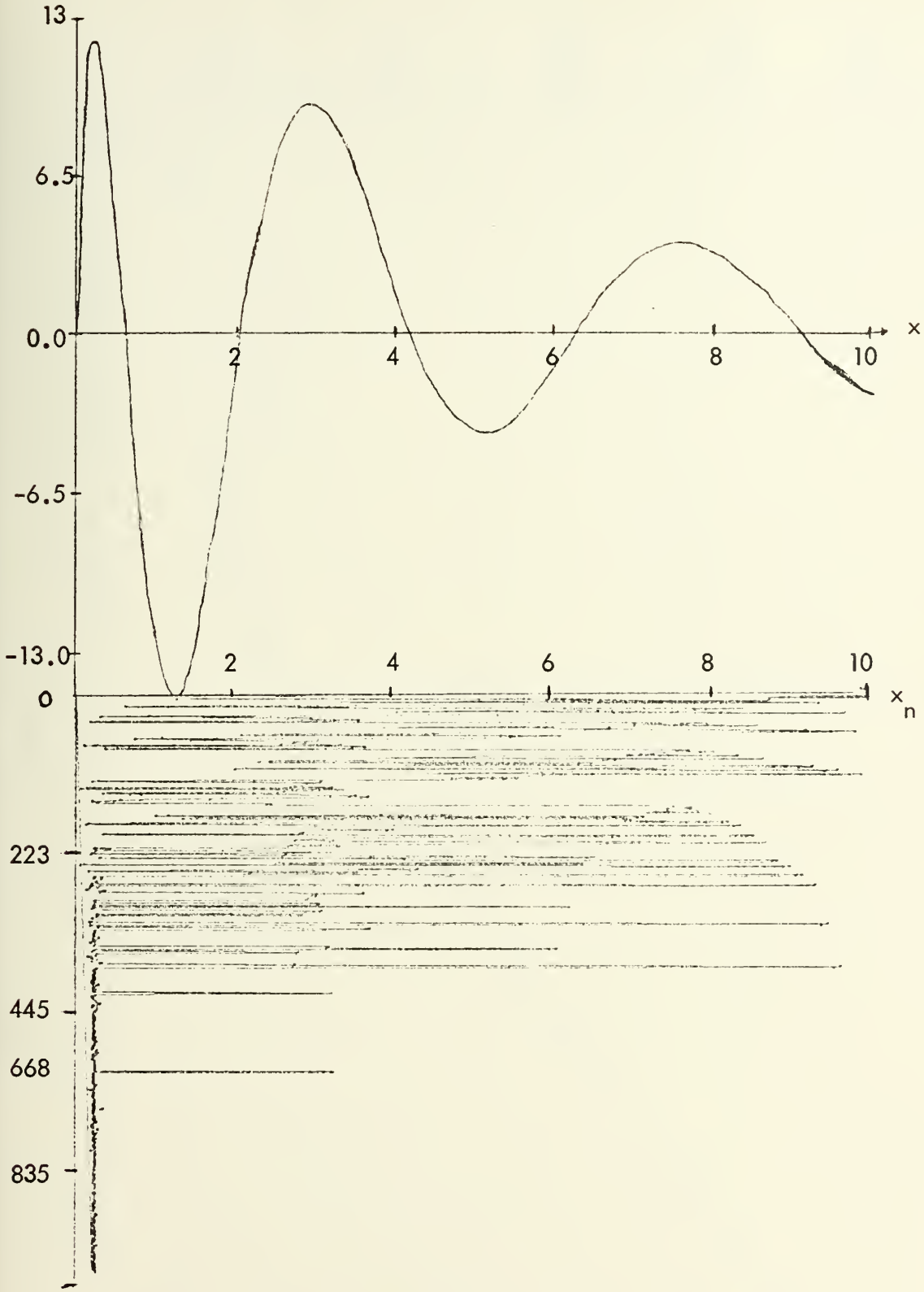


Figure 17. x_n vs. n.

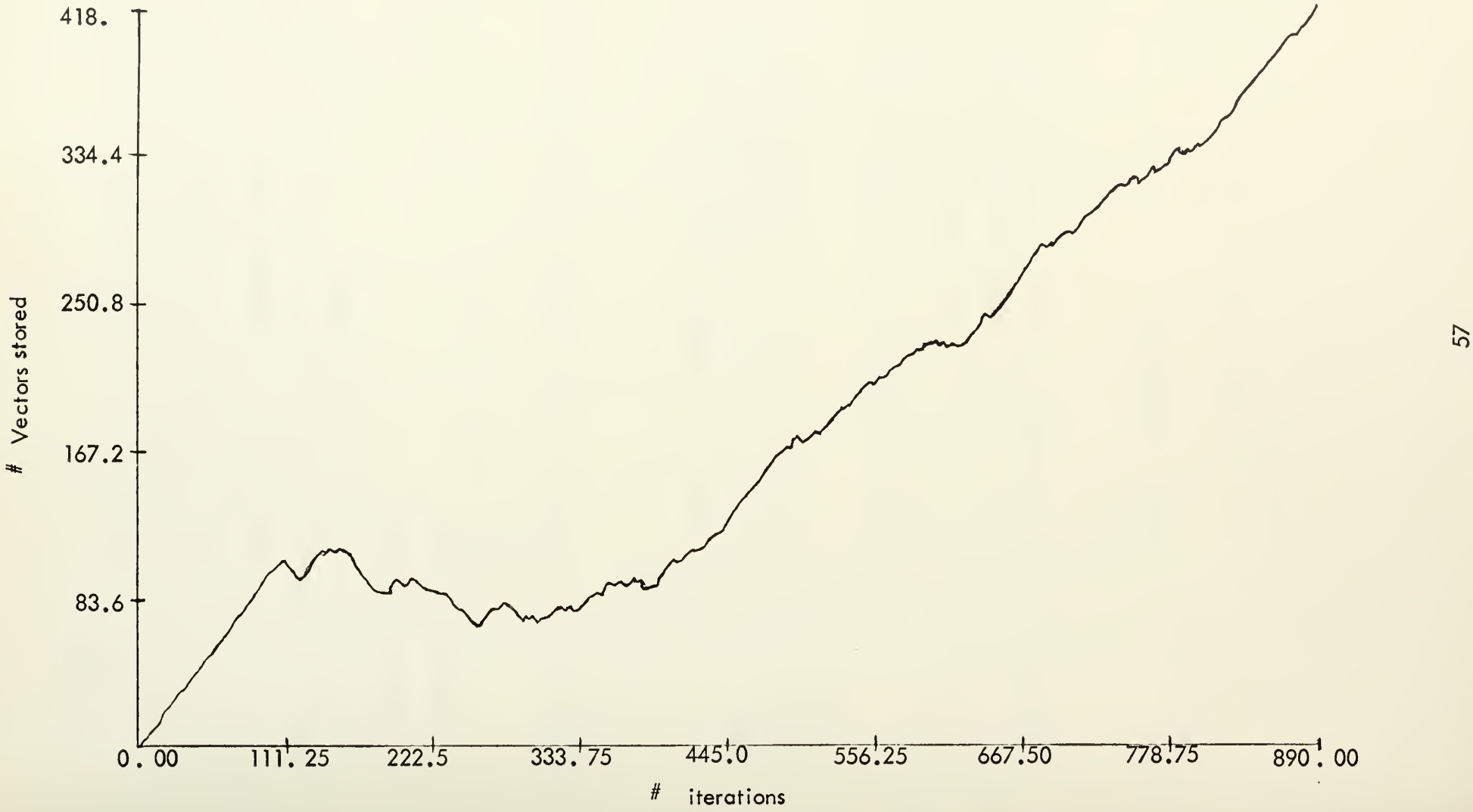


Figure 18. Number of vectors stored vs. number of iterations.

2. Results

For the desired accuracy of $\epsilon = .01$, the algorithm required 890 iterations giving the estimate $\hat{\varphi}_\epsilon = 12.0312$. Since the number of global maxima was determined a priori from Fig. 16 to be 1, Alternative 1 was again used to estimate the location of φ. x was estimated as 0.2415. The largest number of vectors, (t_i, z_i), stored in the computer's memory was 418 and the total computer time (including the time to compute $f(x_n)$) was 23.96 seconds.

C. SAMPLE PROBLEM III

Given the function

$$f(x) = \sum_{i=1}^{5} i \sin[(i+1)x + i],$$

estimate the global maximum φ and its location in $[-10, 10]$ with a desired accuracy of $\epsilon = .01$.

1. Discussion

So far the method under consideration appears to be successful in maximizing functions with a single global maximum. The function under consideration was used to determine the algorithm's ability to maximize functions with more than one global maximum.

A computer plot of f was used to determine the nature of the function, see Fig. 19. It appears from Fig. 19 that f has more than one global maximum in the experimental region. The function is differentiable, but too difficult to locate the maximum using calculus. Gradient techniques were again applied in the intervals assumed to contain the maximum. These methods located the maximum

$$\varphi = 12.0313 \ldots$$

at $$\Psi = \{ -6.7745 \ldots, 5.7918 \ldots \}.$$

C was computed as

$$C = \max_{x \in [-10, 10]} \{ | i(i+1) | \} = 70.$$

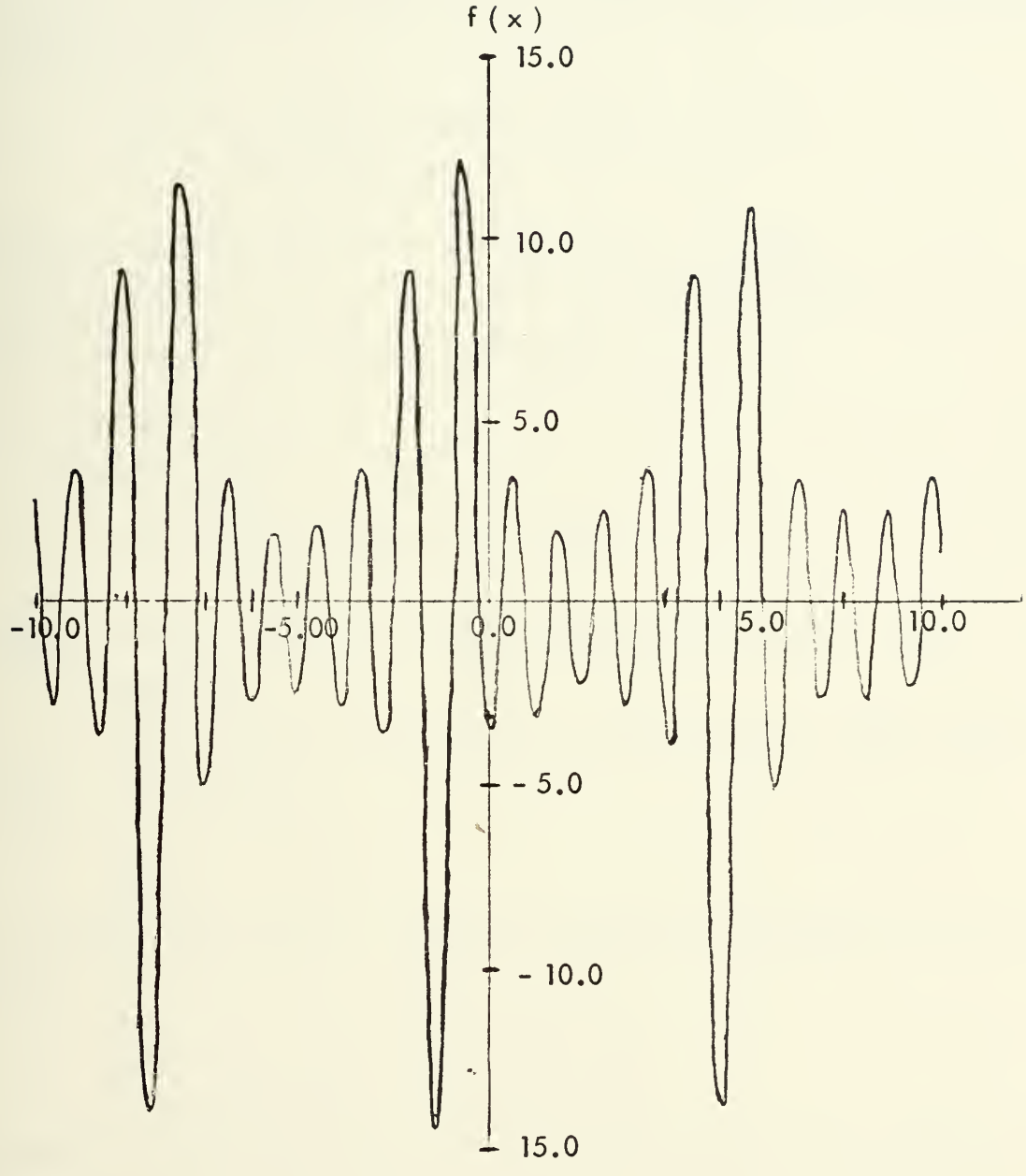


Figure 19. Graph of $f(x) = \sum_{i=1}^{5} i \sin [(i+1)x + i]$.

2. Results

For the desired accuracy of $\varepsilon = .01$, the algorithm required 444 iterations giving the estimate $\hat{\varphi}_{\varepsilon} = 12.0313$. Examination of Fig. 19 revealed the possibility of more than one global maximum in [-10, 10], hence, Alternative IV was used to locate φ in this interval. The residual uncertainty in the location was reduced to three intervals

$$W = \{ [-6.7907, -6.7595], [-0.5129, -0.4261], [5.7749, 5.8061] \}$$

(there is a local maximum $f(x) = 12.0312 \ldots$ at $x = -0.4914 \ldots$.) The largest number of vectors, (t_i, z_i), stored in the computer's memory was less than 250 and the total computer time was 13 seconds.

D. SAMPLE PROBLEM IV

Given the function

$$f(x) = \sum_{i=1}^{5} i \sin [-(i+1)\sqrt{x} + i],$$

estimate the zeros of f in [.01, 10] with a desired accuracy of $\varepsilon = .01$.

1. Discussion

This problem was used to test the algorithm's ability to locate zeros of a function. The function

$$g(x) = -|f(x)|$$

was plotted by the computer in an effort to determine the intervals in [.01, 10] where

$$\max_{x \in [.01, 10]} g(x) = 0$$

(zero will be the global maximum of g as described in Chapter IV.) The location (s) of the global maxima $\varphi = 0$ are the zeros of f. Figure 20 was used to approximate starting points for gradient methods in order to locate the true zeros of f. The zeros were determined as (0.0215 ..., 0.0000), (0.6180 ..., 0.0000), (2.0795 ..., 0.0000), (4.2232 ..., 0.0000), (6.3051 ..., 0.0000),

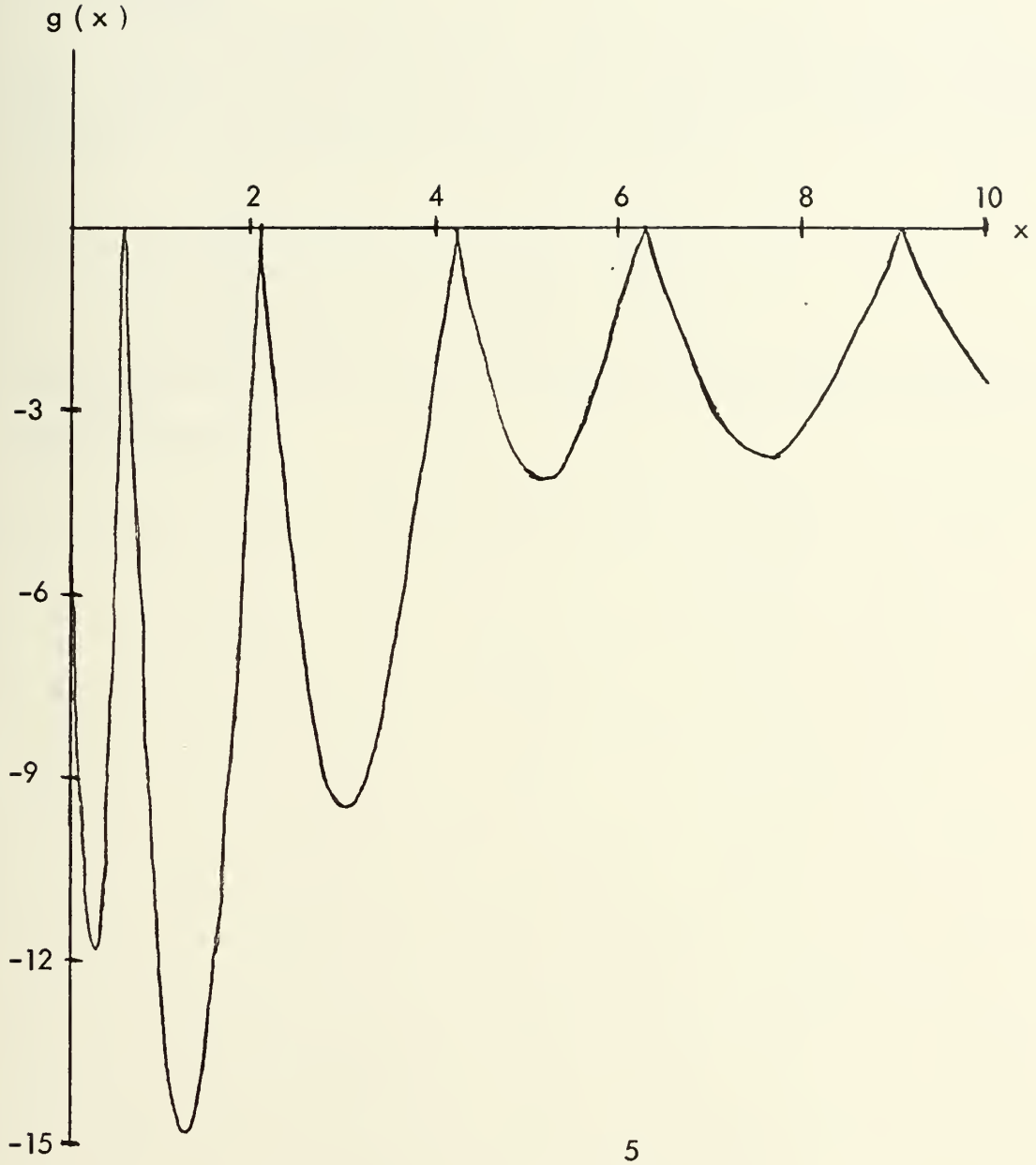


Figure 20. Graph of $g(x) = -\left|\sum_{i=1}^{5} i \sin[-(i+1)x + i]\right|$.

(9.0865 ..., 0.0000), C was computed as

$$C = \max_{x \in [\,.01,\ 10\,]} \left\{ \left| \sum_{i=1}^{5} (i+1)\, x^{-\frac{1}{2}} \right| \right\} = 350.$$

2. Results

For the desired accuracy of $\epsilon = .01$, the algorithm required 2253 iterations. Alternative IV was used to estimate the locations of the zeros. The residual uncertainty in the location was reduced to six intervals

$$[\,0.0214,\ 0.0239\,]\ ,\ [\,0.6173,\ 0.6186\,],$$
$$[\,2.0795,\ 2.1173\,]\ ,\ [\,4.2280\ ,\ 4.4323\,]$$
$$[\,6.2246,\ 7.2566\,]\ ,\ [\,8.8457\ ,\ 9.1452\,]\ .$$

The largest number of vectors (t_i, z_i) stored was 474 and the total computer time was 52.87 seconds.

VI. EXPERIMENT WITH THE COMPUTERIZED ALGORITHM

A. PROCEDURE

The following experiment was performed to test the algorithm's sensitivity to the Lipschitzian constant and the shape of the criterion function. To obtain data for the experiment, eighteen functions of the form

$$f_{\alpha,\beta,\gamma}(x) = \begin{cases} \max\left\{10\left[1 - \left(\frac{x-5}{\beta}\right)^2\right]\right\} & \text{if } 0 \le x \le 10 \\ \max\left\{\alpha\left[1 - \left(\frac{x-5}{\gamma}\right)^2\right]\right\} & \text{if } -10 \le x < 0 \end{cases}$$

were used with the following ranges on the parameters:

$0 \le \alpha \le 10$,

$0 < \beta \le 5$,

$0 < \gamma \le 5$.

Figure 21 illustrates the general form of the function $f_{\alpha,\beta,\gamma}$ and the eighteen functions with their exact parameters are illustrated in Fig. 22.

C was computed as

$$C = \max_{x \in [-10,\, 10]} \left\{ \left| \frac{d f_{\alpha,\beta,\gamma}}{d x} \right| \right\}$$

The Lipschitzian constant C was allowed to vary from its minimum value defined by (17) to five times its minimum value for each of the eighteen functions. The computerized algorithm was used to maximize each variation of $f_{\alpha,\beta,\gamma}$ and the corresponding results; namely, the number of iterations required and the largest number of vectors in D_n stored, were recorded in Table II. All results are based on a desired accuracy of $\epsilon = .01$.

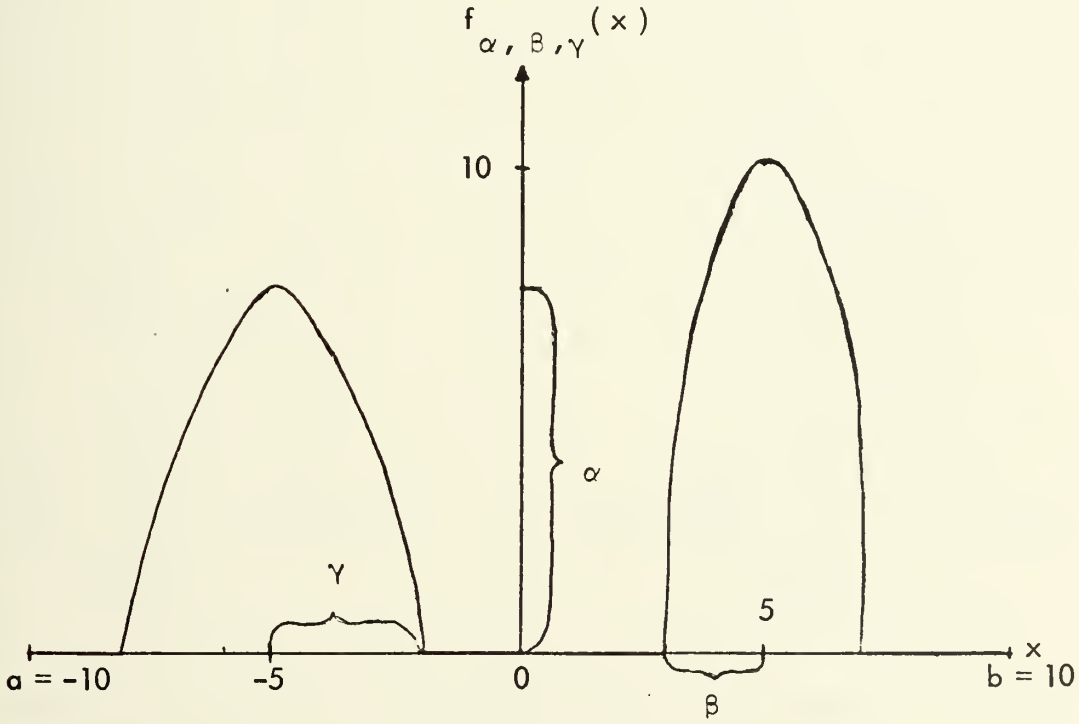
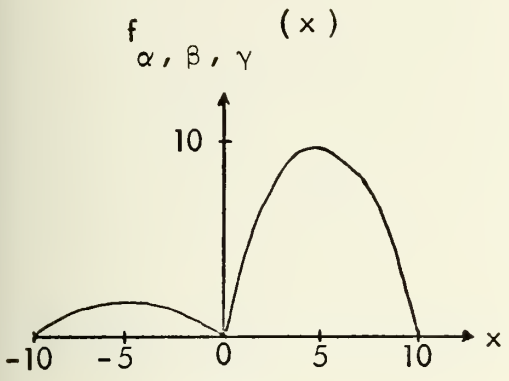
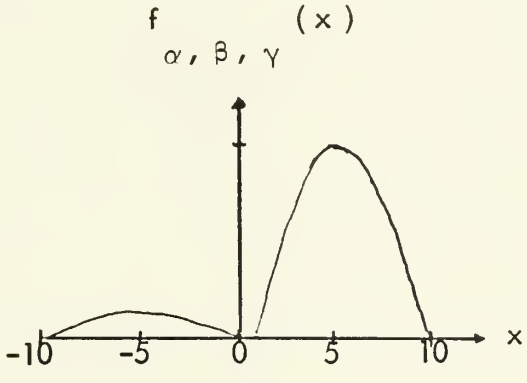


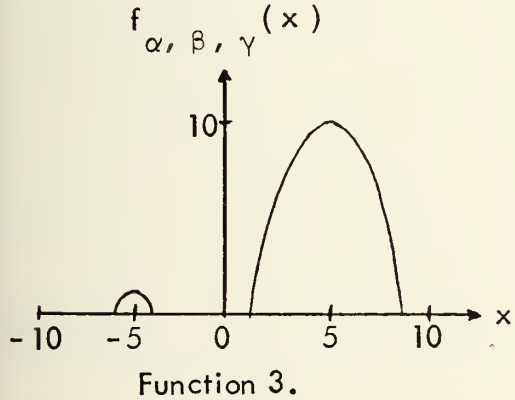
Figure 21. General nature of $f_{\alpha, \beta, \gamma}$



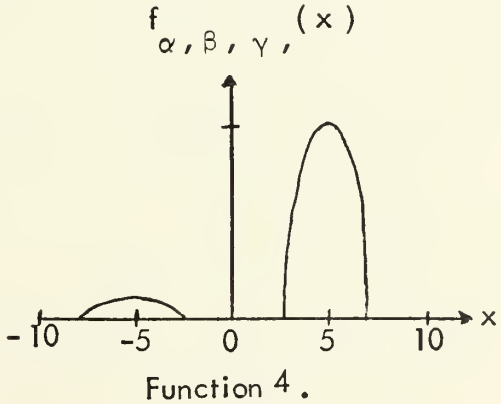
Function 1.
$\alpha = 1, \beta = 1, \gamma = 5$



Function 2.
$\alpha = 1, \beta = 3, \gamma = 4$

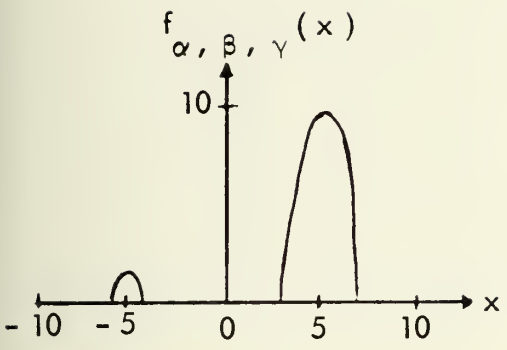


Function 3.
$\alpha = 1, \beta = 4, \gamma = 1$



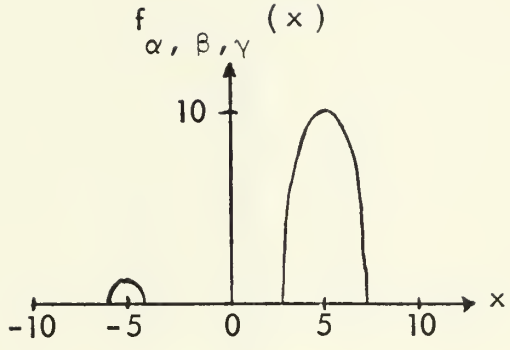
Function 4.
$\alpha = 1, \beta = 2, \gamma = 2$

Figure 22. $f_{\alpha, \beta, \gamma}(x)$ for specified values of shape parameters α, β, γ.



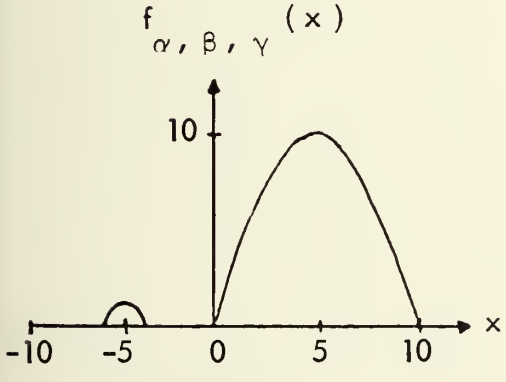
Function 5.

$\alpha = 2, \beta = 2, \gamma = 1$



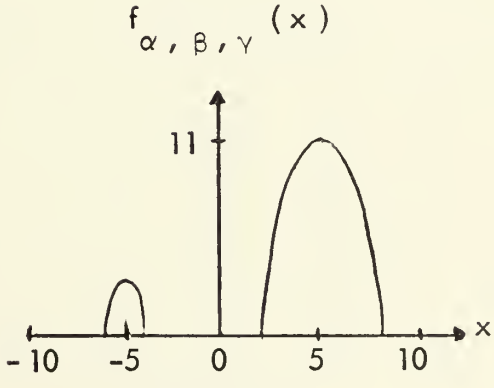
Function 6.

$\alpha = 1, \beta = 2, \gamma = 1$



Function 7.

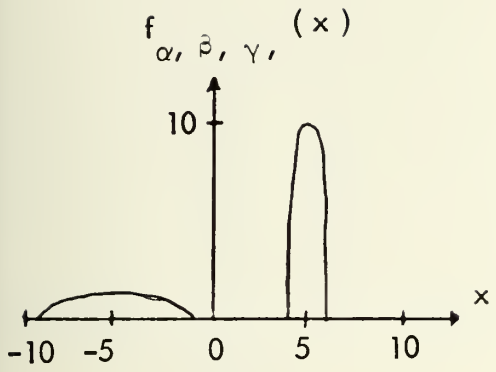
$\alpha = 1, \beta = 5, \gamma = 1$



Function 8.

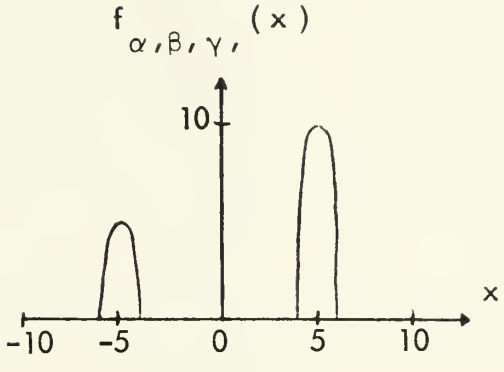
$\alpha = 3, \beta = 3, \gamma = 1$

Figure 22. (Cont.)



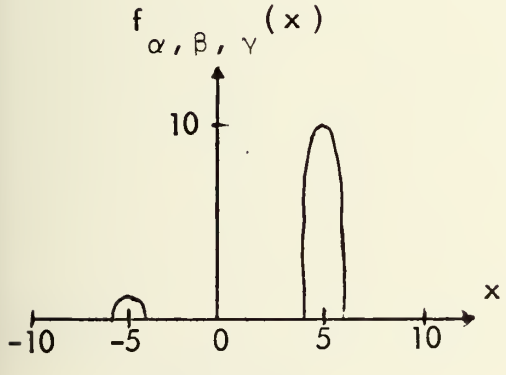
Function 9.

$\alpha = 1, \beta = 1, \gamma = 4$



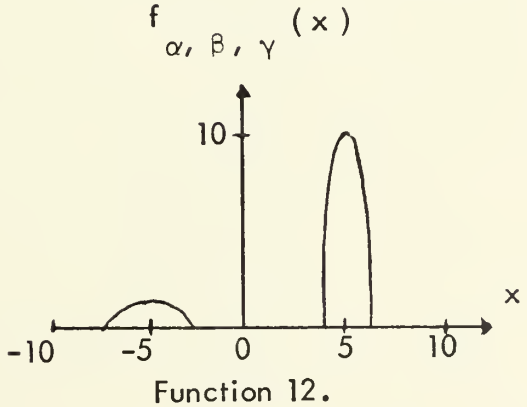
Function 10.

$\alpha = 5, \beta = 1, \gamma = 1$



Function 11.

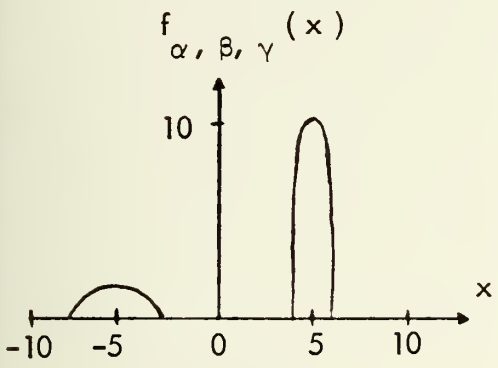
$\alpha = 1, \beta = 1, \gamma = 1$



Function 12.

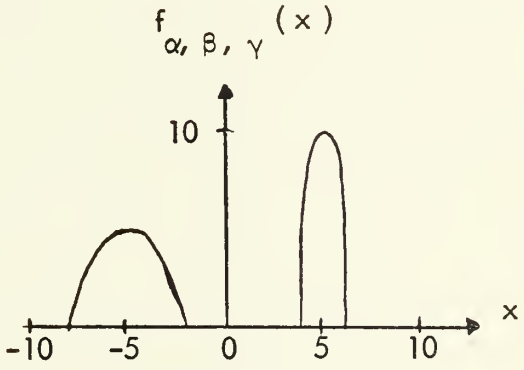
$\alpha = 1, \beta = 1, \gamma = 2$

Figure 22. (Cont.)



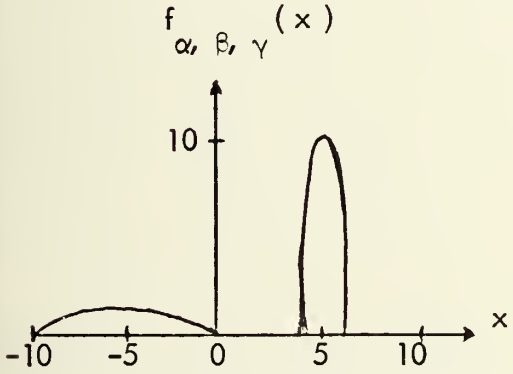
Function 13.

$\alpha = 2, \beta = 1, \gamma = 2$



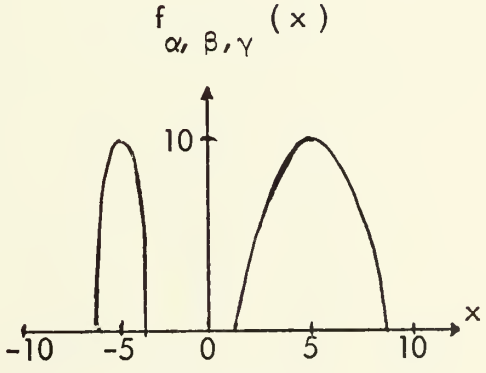
Function 14.

$\alpha = 5, \beta = 1, \gamma = 3$



Function 15.

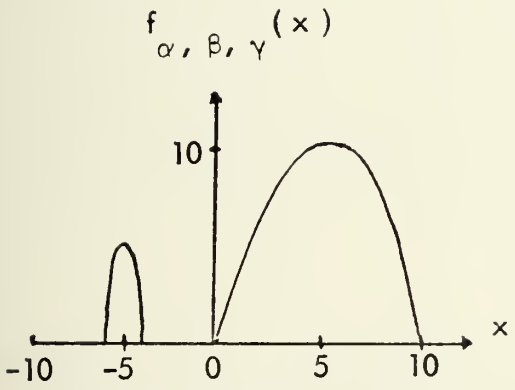
$\alpha = 1, \beta = 1, \gamma = 5$



Function 16.

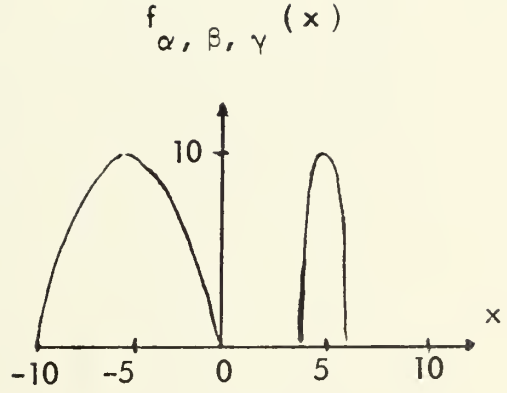
$\alpha = 10, \beta = 4, \gamma = 1$

Figure 22. (Cont.)



Function 17.

$\alpha = 5, \beta = 5, \gamma = 1$



Function 18.

$\alpha = 10, \beta = 1, \gamma = 5$

Figure 22. (Cont.)

	Function #	1	2	3	4	5	6	7	8	9
C_{min}	# Iterations	145	245	293	357	357	357	357	673	833
	# Vectors	84	160	180	208	208	208	232	416	480
$2\ C_{min}$	# Iterations	301	461	557	769	769	769	693	1301	1657
	# Vectors	192	272	332	484	484	484	444	812	936
$3\ C_{min}$	# Iterations	445	741	857	1093	1093	1093	1069	2049	2585
	# Vectors	280	456	552	628	628	628	664	1292	1392
$4\ C_{min}$	# Iterations	569	1001	1193	1533	1533	1533	1477	2561	3313
	# Vectors	372	628	760	936	936	936	960	1572	1852
$5\ C_{min}$	# Iterations	693	1233	1477	1825	1817	1813	1829	3405	4001
	# Vectors	444	768	924	1096	1096	1096	1164	2132	2150
	α	1	1	1	1	2	1	1	3	1
	β	5	3	4	2	2	2	5	3	1
	γ	5	4	1	2	1	1	1	1	4
	C_{min}	4	10	10	25	25	25	10	30	100

Table II. Experimental results for algorithm's sensitivity to the Lipschitzian constant.

	Function #	10	11	12	13	14	15	16	17	18
C_{min}	# Iterations	841	833	833	833	865	833	1537	1833	4321
	# Vectors	480	480	480	480	480	480	960	1164	2784
$2\,C_{min}$	# Iterations	1677	1657	1657	1657	1717	1657	3057	3625	8613
	# Vectors	936	936	936	936	936	936	1872	2304	5492
$3\,C_{min}$	# Iterations	2605	2585	2585	2585	2649	2585	4657	5545	13189
	# Vectors	1392	1392	1392	1392	1392	1392	2784	3570	8560
$4\,C_{min}$	# Iterations	3353	3313	3313	3313	3429	3313	6113	7233	
	# Vectors	1852	1852	1852	1852	1852	1852	3704	4556	
$5\,C_{min}$	# Iterations	3877	3873	3913	3921	3977	4049	7137	8493	
	# Vectors	2146	2146	2150	2146	2146	2146	4292	5300	
	α	5	1	1	2	5	1	10	5	10
	β	1	1	1	1	1	1	4	5	1
	γ	1	1	2	2	3	5	1	1	5
	C_{min}	100	100	100	100	100	100	100	50	100

Table II. (Cont.)

B. EXPERIMENTAL RESULTS

A brief description of Table II follows:

$$C_{min} = \max_{x \in [-10,\ 10]} \left\{ \frac{100}{\beta^2},\ \frac{10\alpha}{\gamma^2} \right\} \tag{18}$$

defined by (17). $i\ C_{min}$, $i = 1, 2, 3, 4, 5$ were the variations of C used by the algorithm to estimate the global maximum φ for the functions 1 - 18 specified by the shape parameters α, β, γ. Figure 22 illustrates the eighteen functions under consideration. The data points are the number of iterations required and the largest number of vectors in D_n for the corresponding $i\ C_{min}$ and function.

The columns of Table II and the graphs of the tabulated results, Fig. 23 and Fig. 24, seem to confirm the theoretical conjecture that the number of iterations and the number of vectors stored increase approximately in direct proportion to $i\ C_{min}$.

The effects of the shape of the criterion function are not so readily apparent from Table II and Figs. 23 and 24. In order to obtain these results it is necessary to determine the value of the data points for each function with different shape parameters evaluated for the same constant C. From Table II it is obvious that the common C would have to be 100 since

$$\max C_{min} = 100 .$$

Unfortunately time was not available to compute the data points for each function using a common C. In an effort to use the available data to estimate the algorithm's sensitivity to function shape, it was hypothesized that for $i = 1$

$$\frac{\max C_{min}}{C_{min}} \cdot I ,$$

where I is the number of iterations required when $C = C_{min}$, would approximately equal the number of iterations required if $C_{min} = \max C_{min}$. Similarly, for $i = 1$, it was hypothesized that

$$\frac{\max C_{min}}{C_{min}} \cdot S ,$$

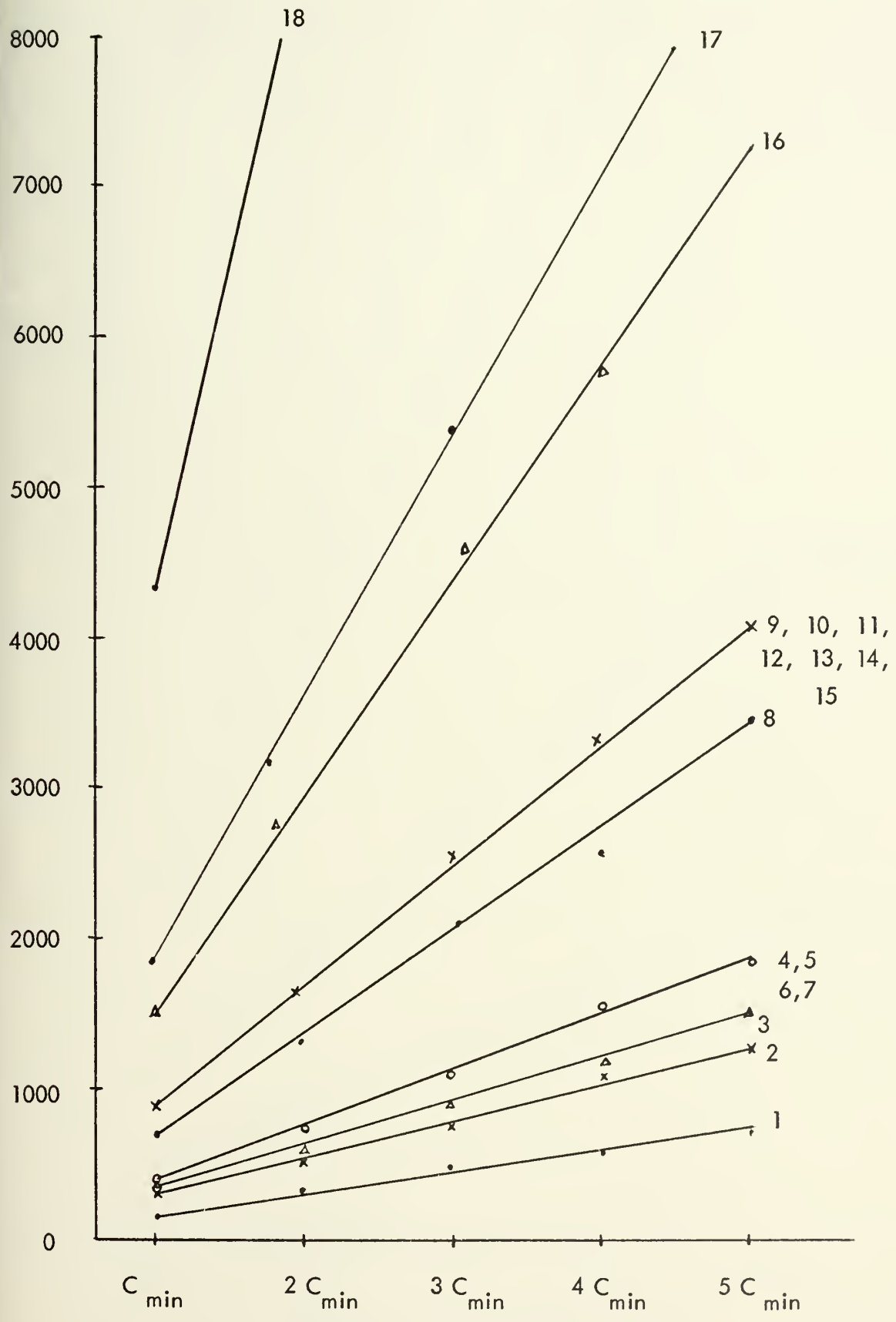


Figure 23. Iterations required vs. i C_{min}.

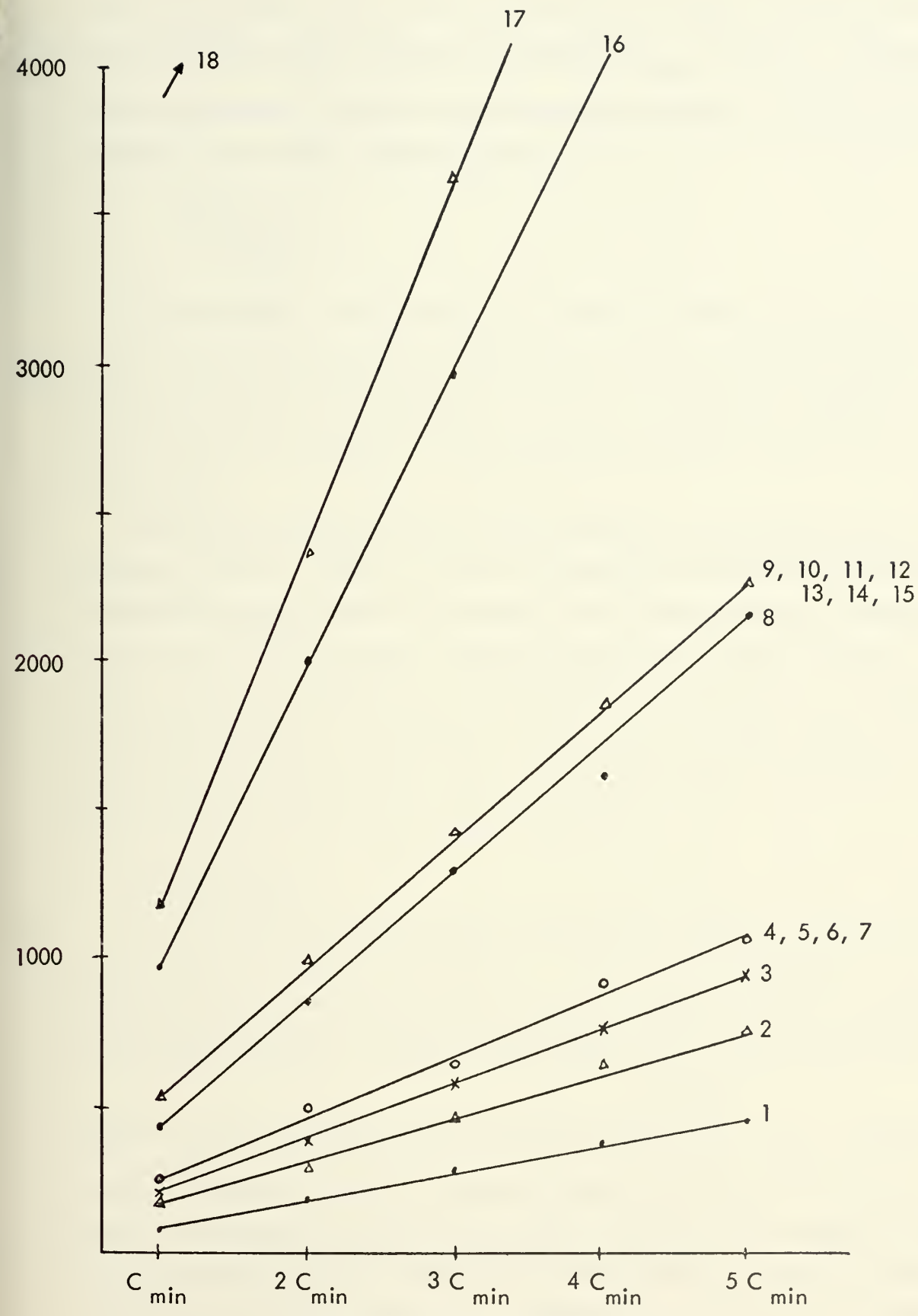


Figure 24. Vectors stored vs. i C_{min}.

where S is the largest number of vectors in D_n when $C = C_{min}$, would approximately equal the largest number of vectors in D_n required if $C_{min} = \max C_{min}$. This hypothesis seems plausible since both iterations required and amount of vector storage required is approximately in direct proportion to

$$\frac{\max C_{min}}{C_{min}} .$$

The results based on this hypothesis are tabulated in Table III. Even from this table it is not readily apparent how the shape of the function affects the algorithm, due to the interaction of the parameters. However, the data points obtained for each function can be compared to their corresponding graphs in Fig. 22 to analyze these effects.

The data points of Table III have been rearranged in nonincreasing order and tabulated in Table IV. Table IV makes the comparisons between data points and corresponding figures a little more systematic. From these comparisons it becomes obvious that the largest number of iterations required and the largest number of vectors required to be stored occur when the function is relatively flat in the area of the global maximum φ. Furthermore, one can see by a similar comparison that as other local maxima less than or equal to the global maximum φ get closer to φ the iterations and vectors in D_n increase accordingly.

C. CONCLUSIONS

It was concluded from experimental results that the algorithm is most sensitive to the value of the Lipschitzian constant C. If the value of C selected by the experimenter (call this value C_s) is greater than C_{min} defined by (18) then the search for the global maximum φ will require approximately $\left(\frac{C_s}{C_{min}}\right)$ times as many iterations and vector storage required than would be required if $C_s = C_{min}$. Furthermore, as the number of iterations increase the amount of computer time necessary to make the computations increase proportionately. Similarly, as the number of vectors in D_n increase the amount of computer core memory necessary to store the vectors increases proportionately. Hence, since computer cost is a function of time and

Function #	1	2	3	4	5	6	7	8	9
# Iterations	3625	2450	2930	1428	1428	1428	3570	2243	833
# Vectors	2100	1600	1800	832	832	832	2320	1386	480

Function #	10	11	12	13	14	15	16	17	18
# Iterations	841	833	833	833	865	833	1537	3666	4321
# Vectors	480	480	480	480	480	480	960	2328	2784

Table III. Hypothesized effects of the algorithm's sensitivity to the function shape.

Function #	18	17	1	7	3	2	8	16	4
# Iterations	4321	3666	3625	3570	2930	2450	2243	1537	1428
# Vectors	2784	2328	2100	2320	1800	1600	1386	960	832

Function #	5	6	14	10	9	11	12	13	15
# Iterations	1428	1428	865	841	833	833	833	833	833
# Vectors	832	832	480	480	480	480	480	480	480

Table IV. Rearrangement of iterations/vectors for ease of comparison.

core required, the cost of the experiment will increase proportional to C_s. Clearly, experimental costs will be optimized when $C_s = C_{min}$.

If the value for C cannot be obtained analytically and the nature of the function is unknown, the experimenter must be cautious in his selection of C_s. If $C_s < C_{min}$ it is possible that the algorithm will fail completely. On the other hand the algorithm will always work if $C_s > C_{min}$, but at a higher cost. Therefore, if the exact value for C_{min} cannot be determined a priori, it would be better for the experimenter to risk an over estimate of C at a greater cost than to underestimate and accomplish nothing.

It can also be concluded from the experimental results that the algorithm is sensitive to the function's shape, though not as sensitive as it is to the Lipschitzian constant. In regards to shape, the algorithm is most sensitive to how flat the function is in the area of the global maximum. Of secondary importance, but still significant is the height of all local maxima, in the experimental region, less than the value of φ. As the heights of these local maxima approach φ, the computation cost increases accordingly. Furthermore, though not established by experimental results, it seems plausible that as the number of local maxima relatively close to φ increase the cost will increase accordingly.

Knowledge of the shape of the function to be maximized may be useful for making gross cost estimates or as a programming aide.

VII. RESULTS

A. RESULTS OF SAMPLE PROBLEMS

Consider Sample Problem III of Chapter V. The grid search would require

$$p = \frac{d}{\delta_N}$$

sampling points for the same accuracy of $\epsilon = .01$, where $d = |(-10) - (10)| = 20$ and

$$\delta_N = \frac{2\epsilon}{C} = \frac{.02}{70} = .000\,285\ldots$$

Thus, the number of sampling points required by the grid search is approximately

$$p = 7 \times 10^4$$

while the number of sampling points required by the sequential method proposed by this research was less than 250. Similar comparisons can be made for the other sample problems and the results would be relatively the same, namely, the number of sampling points required would be considerably higher for grid search than for the sequential method proposed by this research.

Purely random methods would require about twice as many sampling points as the grid search for the same accuracy and for a sufficient confidence level.

This indicates that for nontrivial functions, the convergence rate of the algorithm tends to be much faster than the slowest theoretical rate $\frac{C(b-a)}{n}$.

B. CONSEQUENCES OF MEASUREMENTS ERRORS OR SMALLER C THAN REQUIRED

Figure 25 illustrates the consequence of a measurement error, in this case, when the measured value of the function is less than the actual value in the region containing the maximum. Figures 26 and 27 illustrate the consequences of a smaller C than required by the algorithm.

In both cases $F_n(x)$ may fail to be an upperbound to $f(x)$ in some regions of the domain $[a, b]$. Hence, some regions may be excluded from future search. If these regions happen to contain points where the global maximum is attained, the

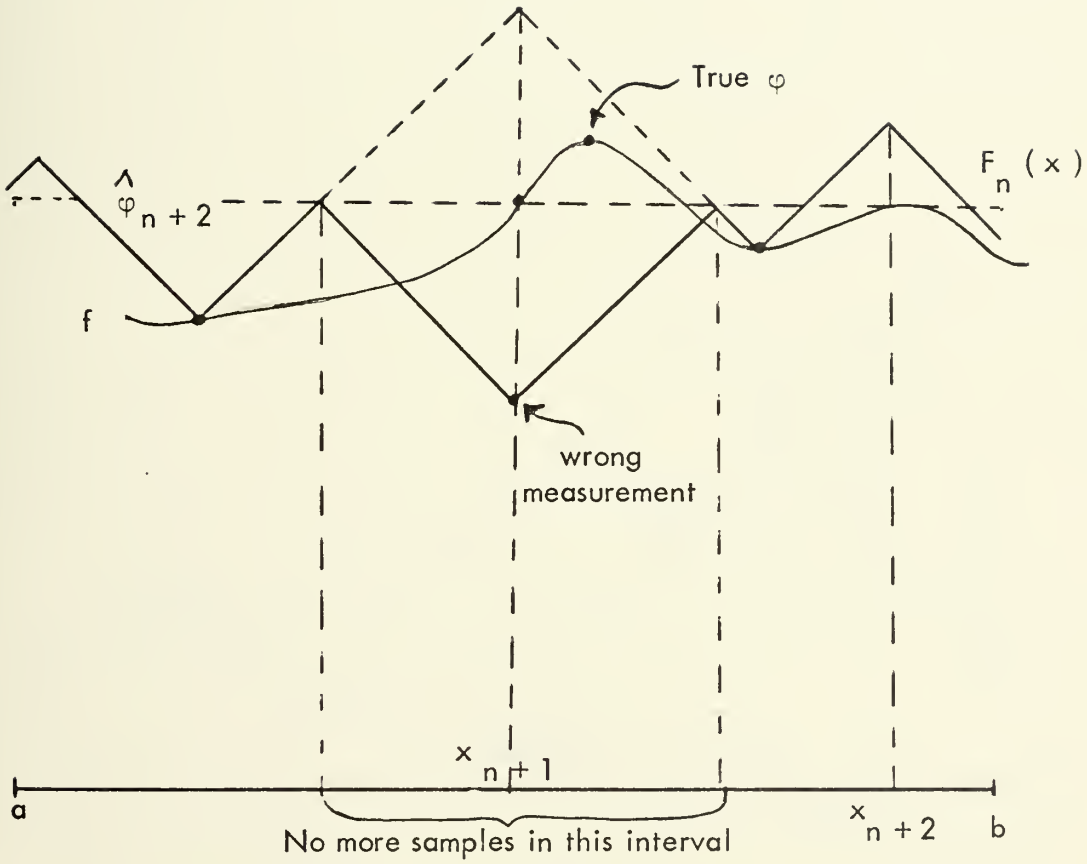


Figure 25. Consequences of measurement errors.

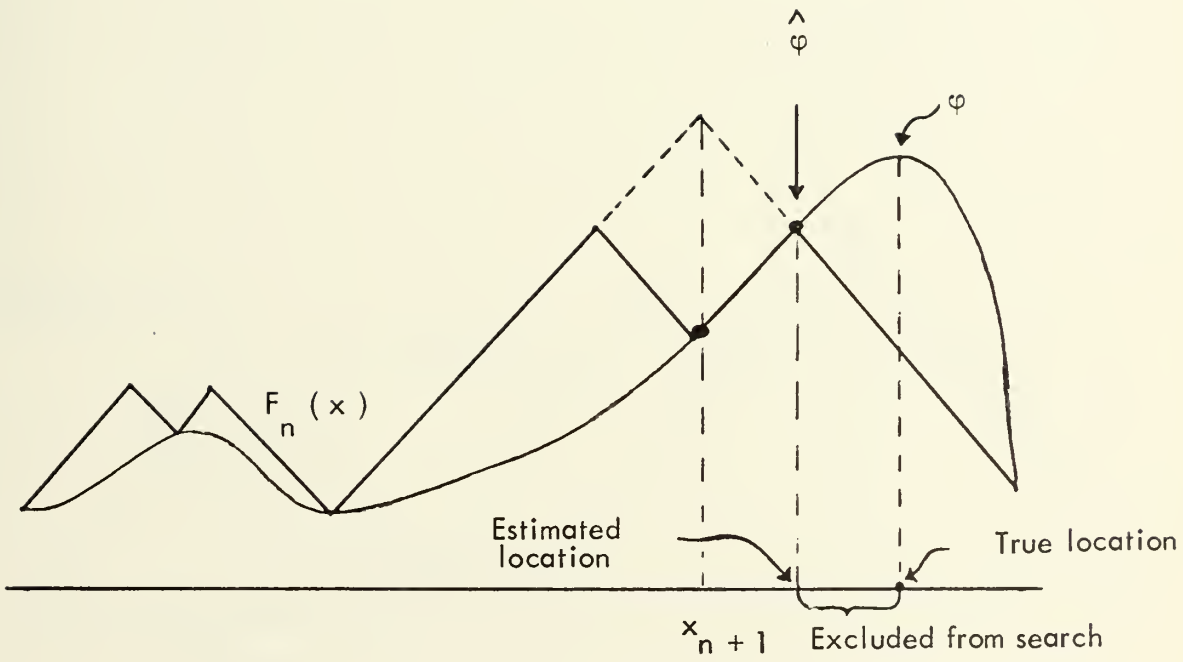


Figure 26. Case where smaller C than required is not recognized by data set.

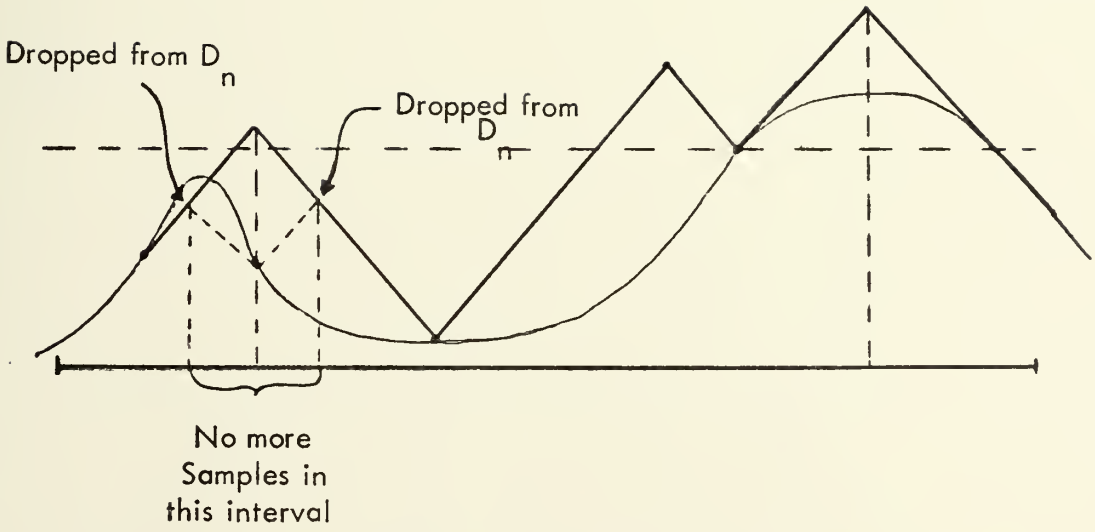


Figure 27. Case where C is underestimated but has no effect on algorithm.

algorithm will result in error, see Figs. 25 and 26. In both of the cases illustrated, (Figs. 25, 26), the value of φ will be underestimated and the location will be wrong. On the other hand it is possible for the algorithm to be successful even if C is less than required, see Fig. 27. If the portion of the function that requires C to be large lies considerably below the global maximum it is still possible to obtain desired results from the algorithm.

The errors may be detected in some cases by obtaining

$$f(t_{H_n}) > z_{H_n}$$

at some point in the sampling sequence, however, this need not necessarily happen, see Fig. 26.

VIII. CONCLUSIONS

The results from the comparisons made between the sequential method proposed by this research and the nonsequential methods of grid and random search clearly indicate that the proposed sequential method is preferable to the nonsequential methods, especially in cases where the function to be maximized is difficult or time consuming to evaluate or when the sampling of a function is costly.

The primary advantage the proposed sequential maximization method has over other methods that are sequential, such as gradient and min-max methods, is that the proposed method is not restricted by the modality of the function, nor does it rely on the regulairty conditions which must be satisfied by these other sequential methods.

IX. RECOMMENDATIONS FOR FUTURE RESEARCH

The sequential search procedure proposed by this research has proved successful in estimating the global maximum and its location of a function whose exact form in the experimental region was known. It is obvious that the same method could be applied to functions whose exact form is unknown. It is the recommendation of the author that future research be conducted in this area by investigating practical examples of the following type:

Consider the "black box" case, where the output of the box is a function of a single independent input variable ranging over the experimental region. The selection of the Lipschitzian constant may be difficult in this situation but may be obtained, by the experimenter's previous knowledge of the system under investigation or from experts who are familiar with the system. By data linking the input/output signals of the box to the on line computerized algorithm the same results can be obtained. See Fig. 28 for an illustration of this approach.

The algorithm proposed by this research could also be experimentally investigated in light of maximizing discrete functions, which may lead to several interesting practical problems.

In principle, it appears that the method could be extended to functions of several variables, but at present it does not seem to be computationally feasible.

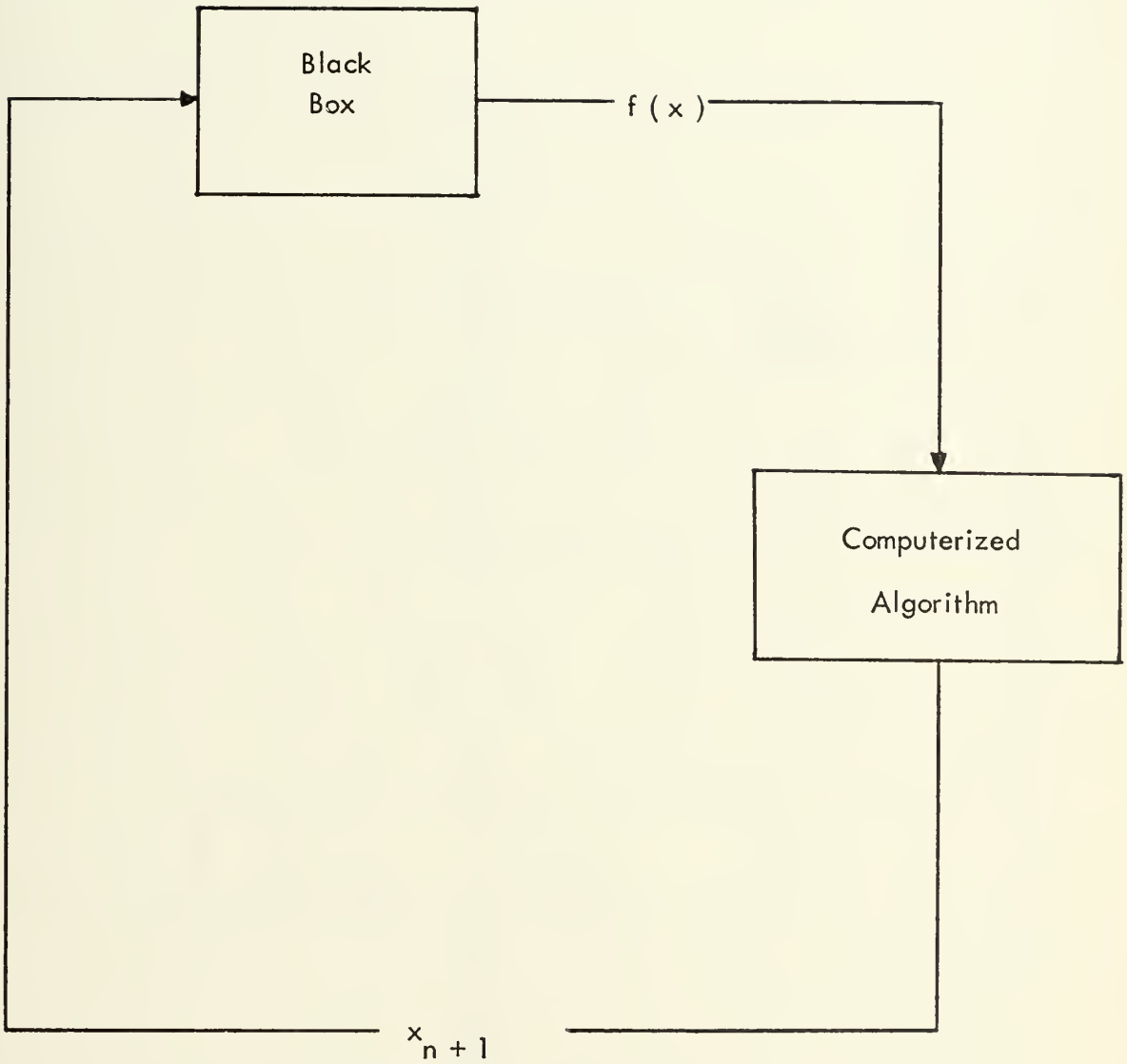
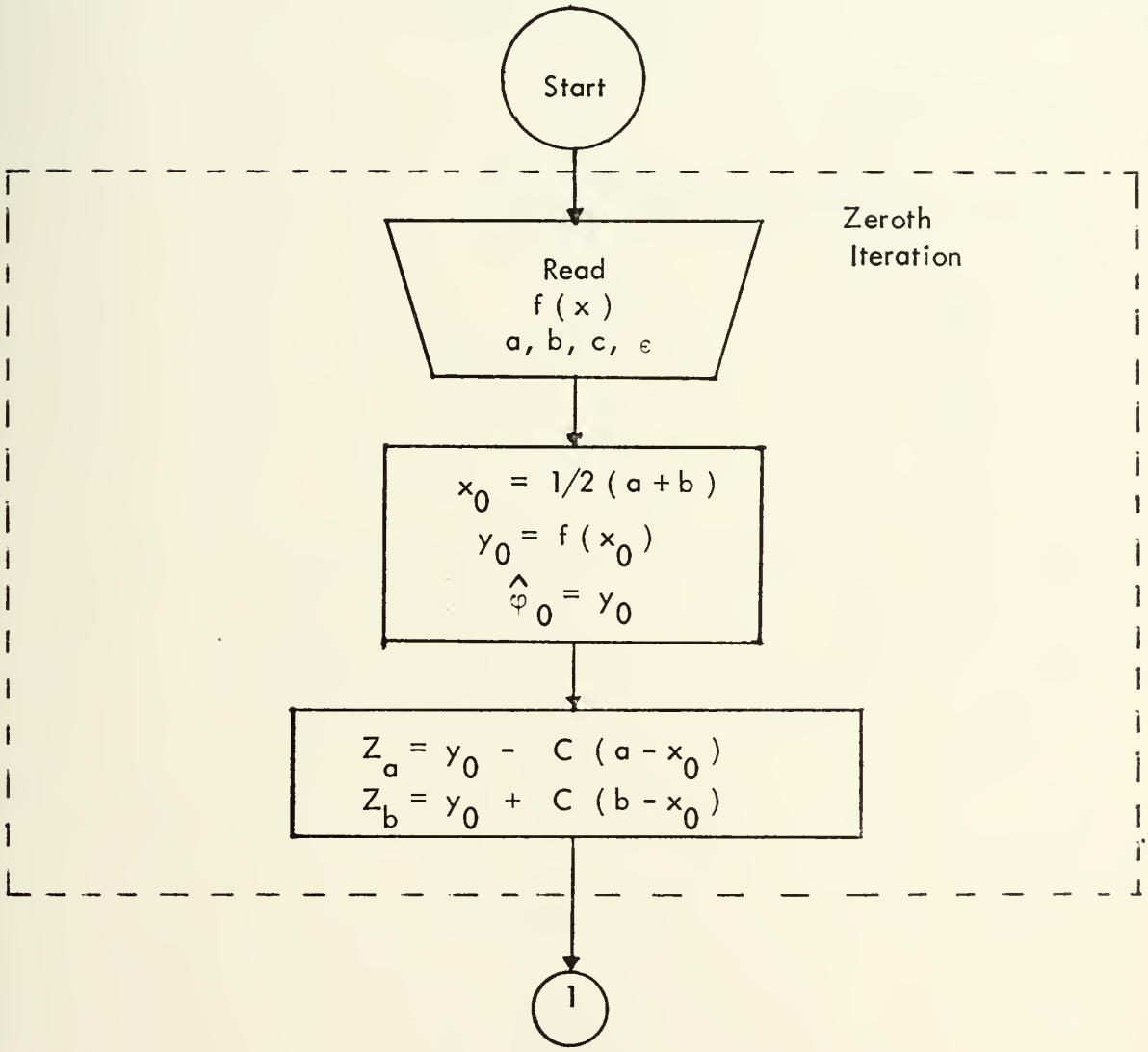
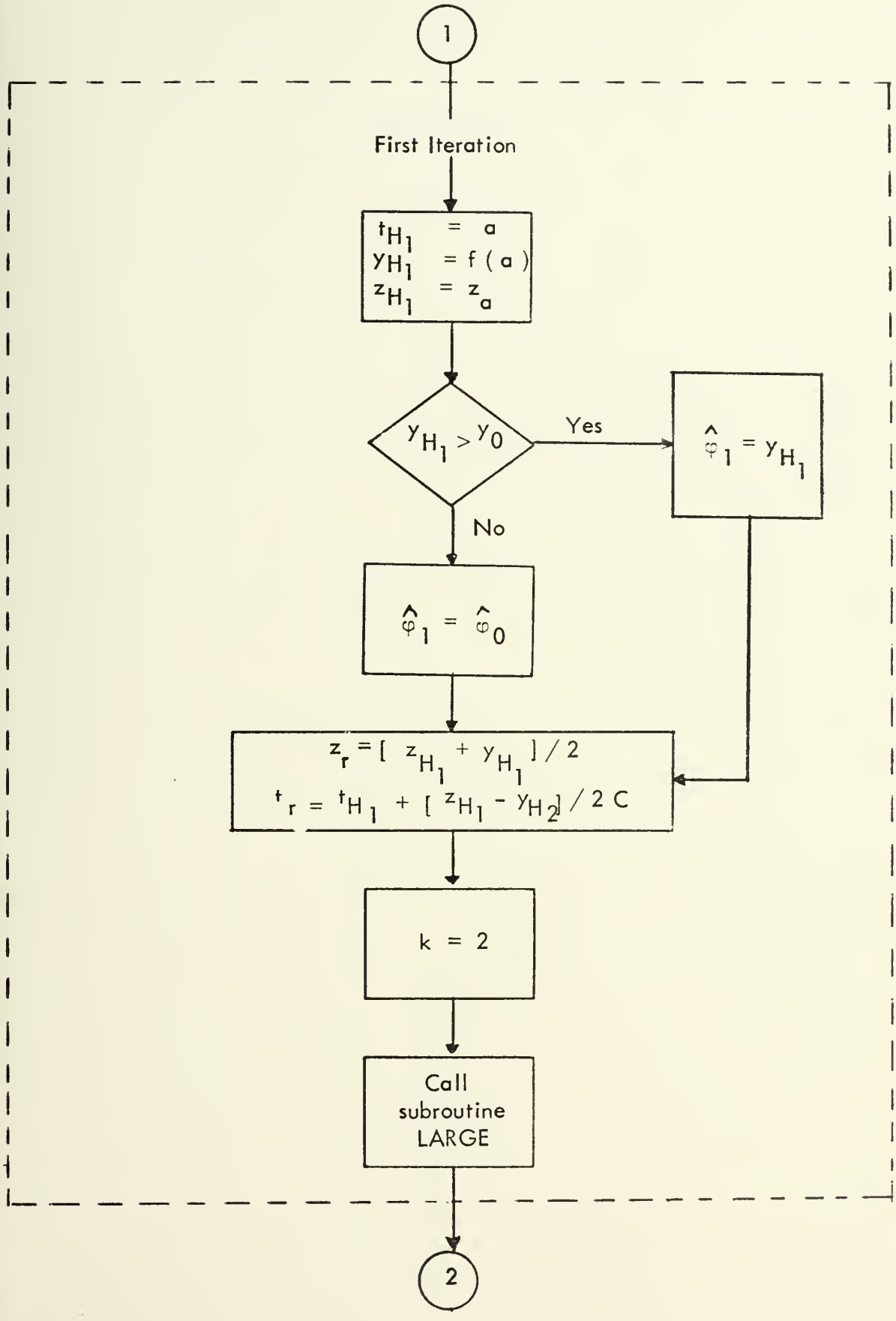


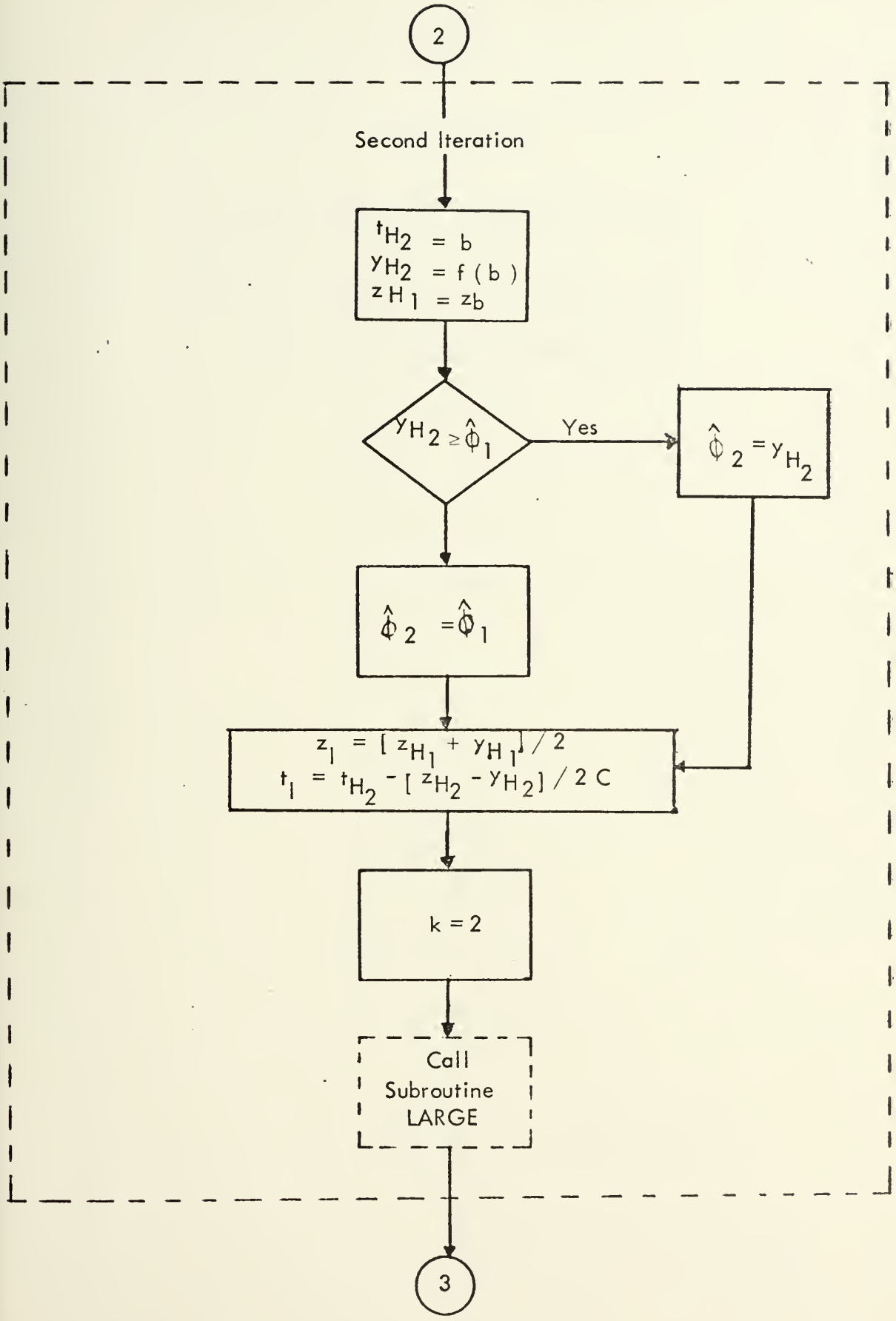
Figure 28. Illustration of "black box" case.

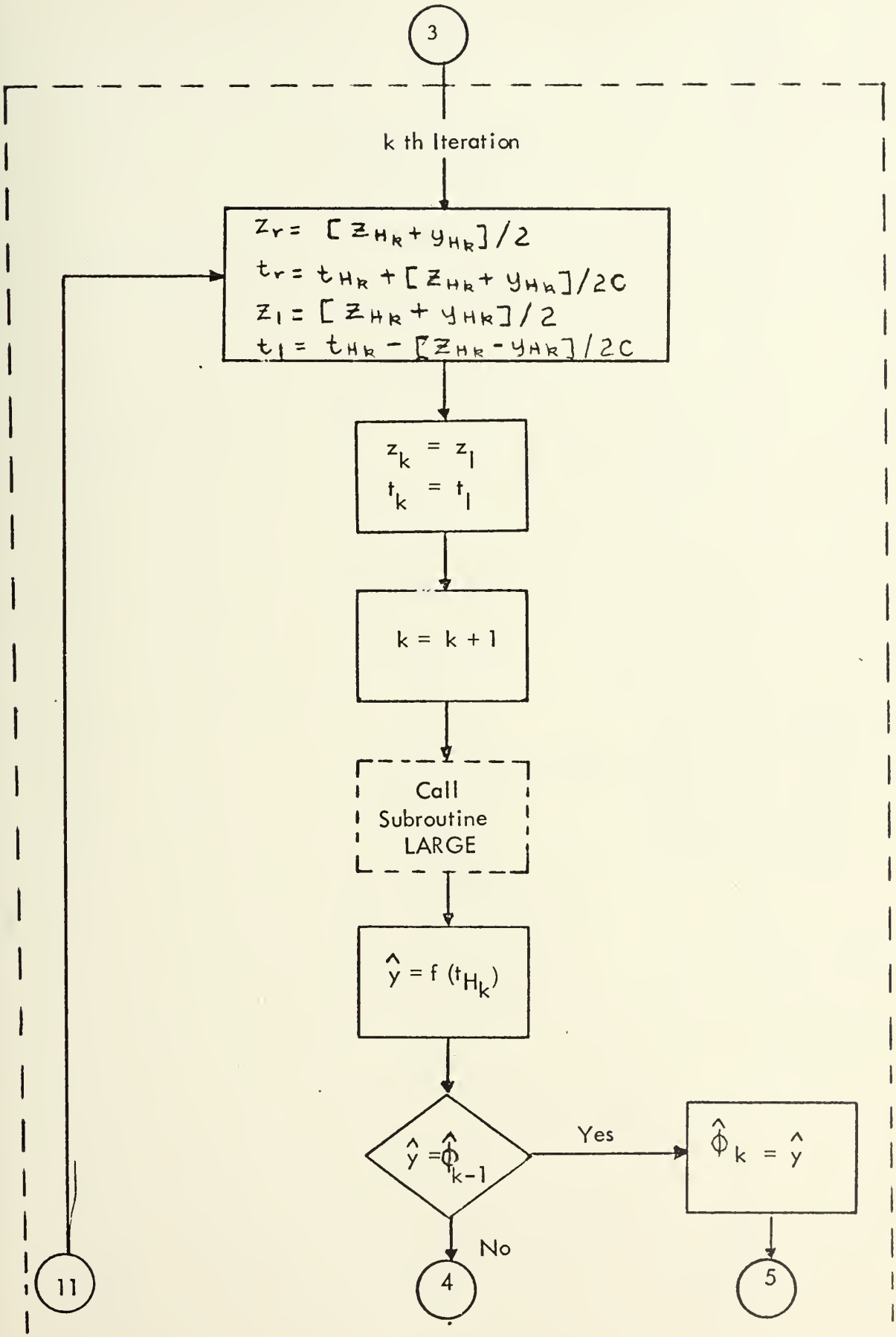
APPENDIX A

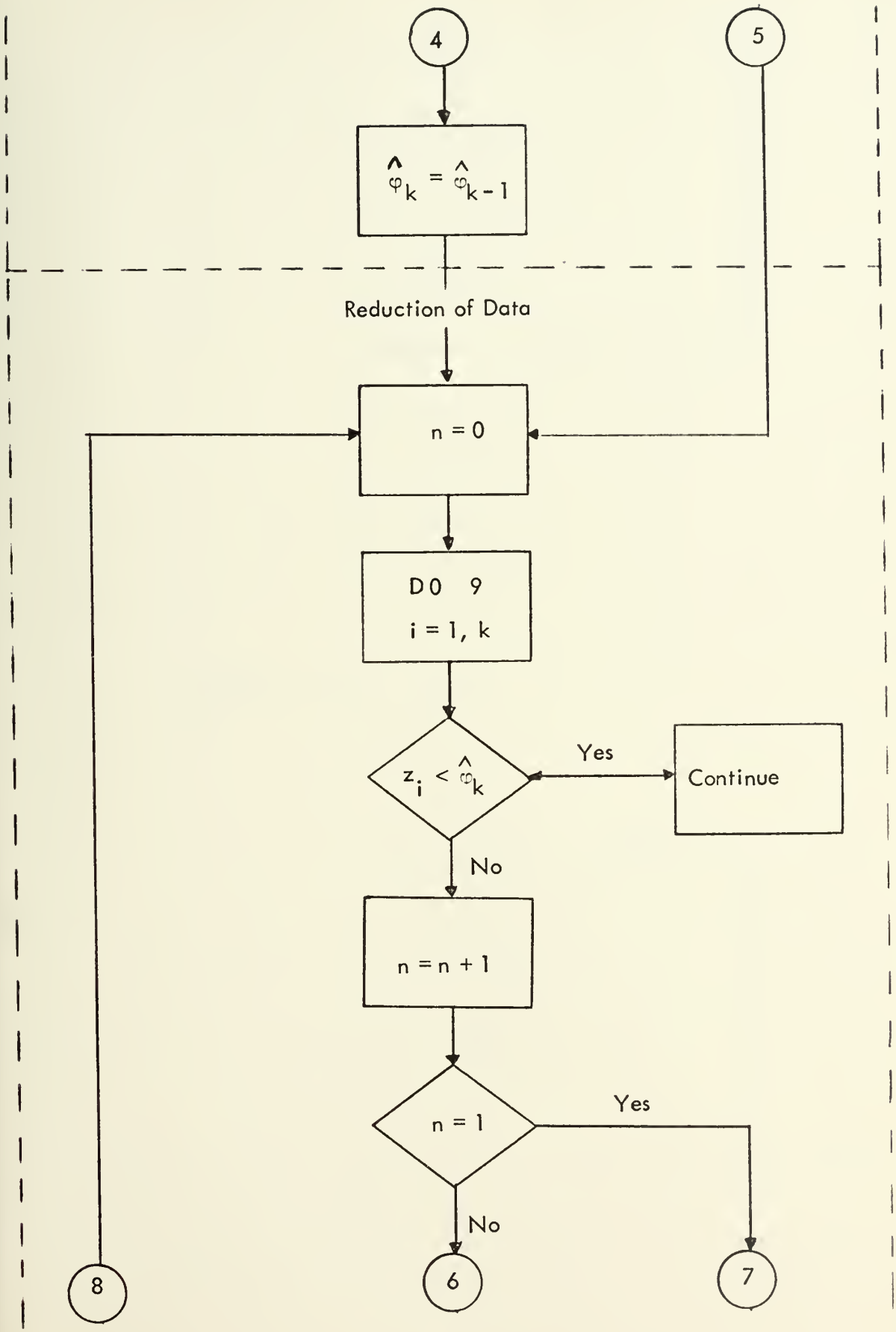
FLOW CHART FOR COMPUTER PROGRAM
WHICH ESTIMATES THE GLOBAL MAXIMUM

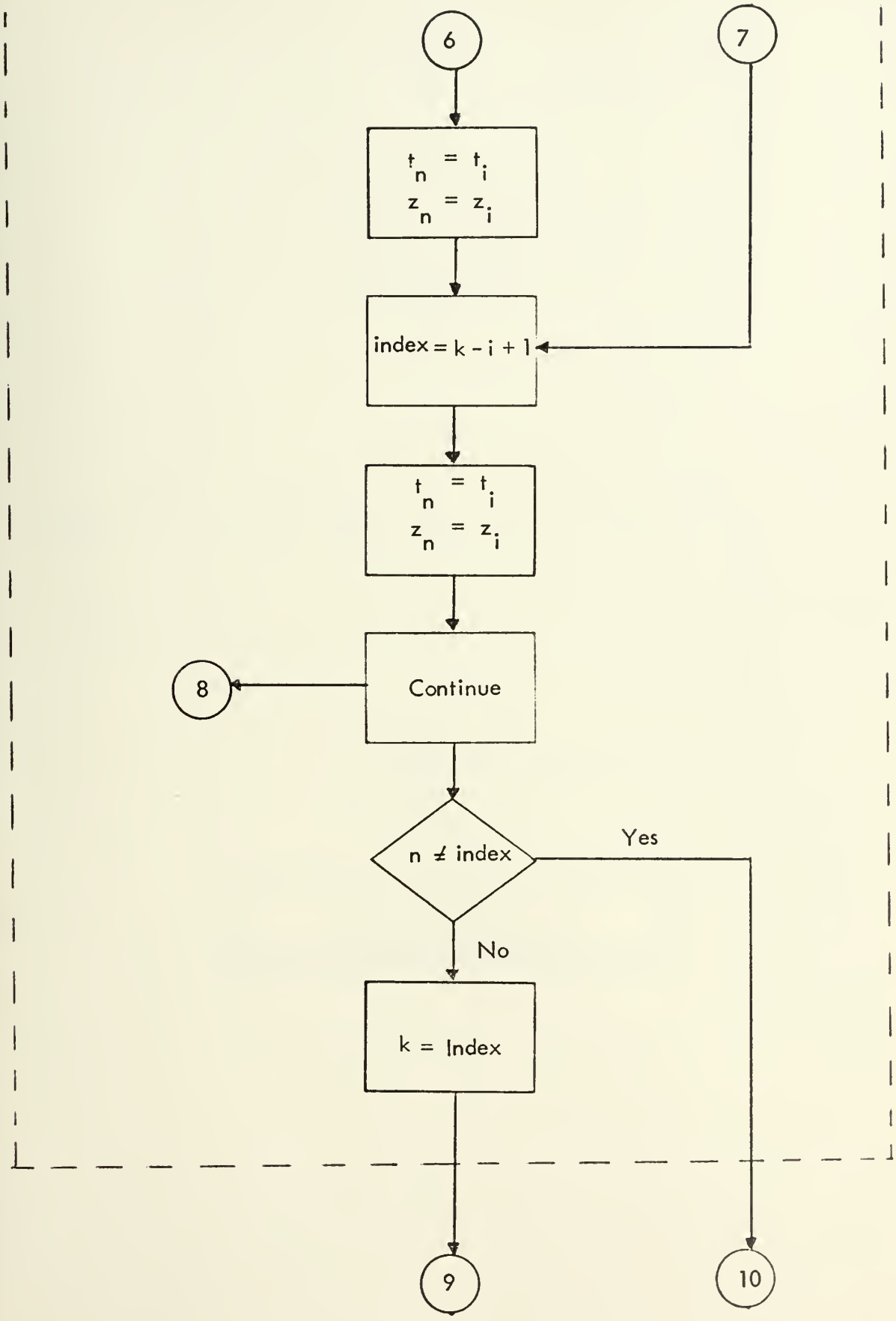


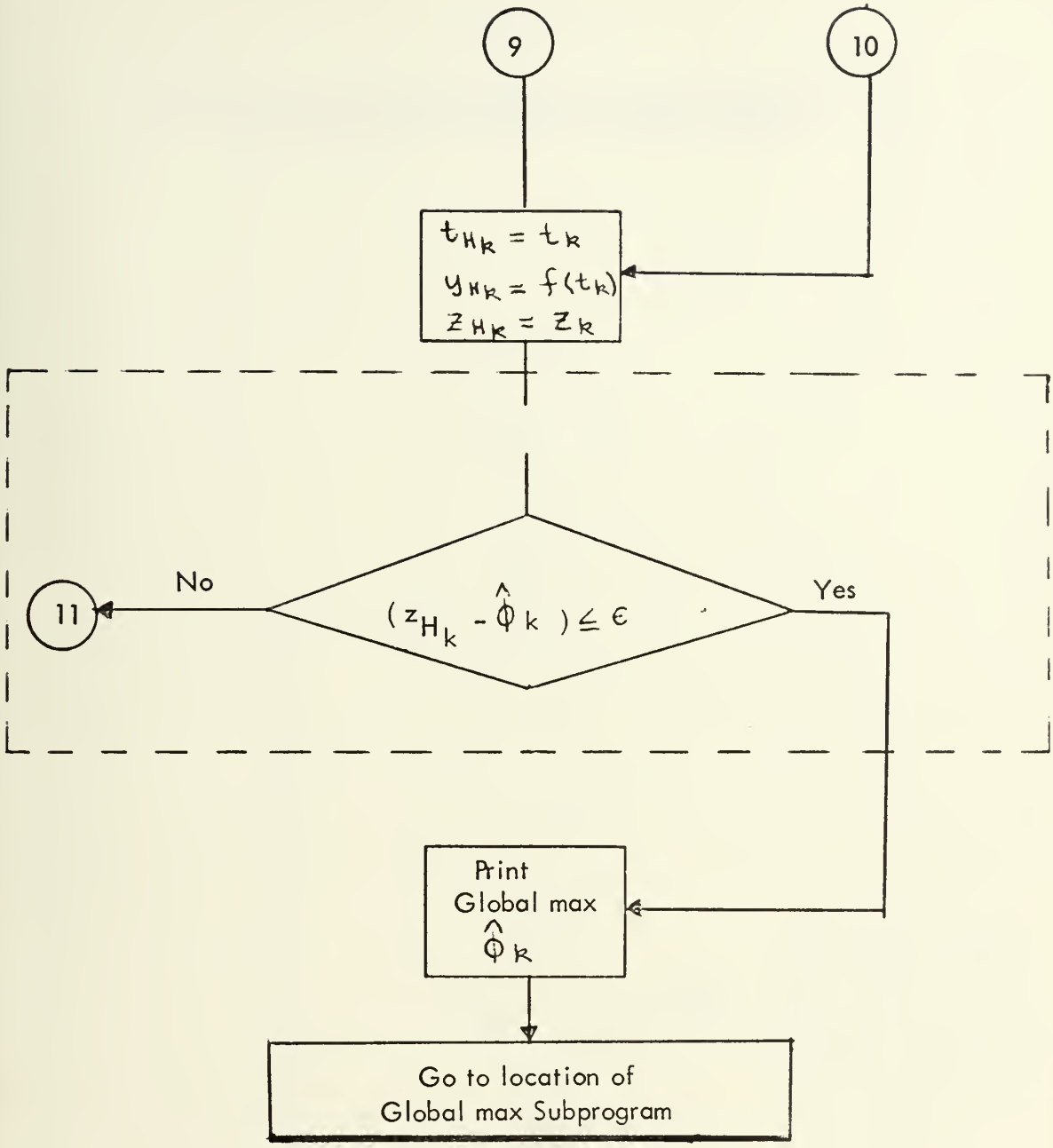

1
First Iteration
$t_{H_1} = a$
$y_{H_1} = f(a)$
$z_{H_1} = z_a$
$y_{H_1} > y_0$
Yes
$\hat{\varphi}_1 = y_{H_1}$
No
$\hat{\varphi}_1 = \hat{\varphi}_0$
$z_r = [z_{H_1} + y_{H_1}] / 2$
$t_r = t_{H_1} + [z_{H_1} - y_{H_2}] / 2C$
k = 2
Call subroutine LARGE
2


2
Second Iteration
$t_{H_2} = b$
$y_{H_2} = f(b)$
$z_{H_1} = z_b$
$y_{H_2} \geq \hat{\Phi}_1$
Yes
$\hat{\Phi}_2 = y_{H_2}$
$\hat{\Phi}_2 = \hat{\Phi}_1$
$z_1 = [z_{H_1} + y_{H_1}] / 2$
$t_1 = t_{H_2} - [z_{H_2} - y_{H_2}] / 2C$
k = 2
Call Subroutine LARGE
3


3
k th Iteration
$z_r = [z_{H_R} + y_{H_R}]/2$
$t_r = t_{H_R} + [z_{H_R} + y_{H_R}]/2C$
$z_l = [z_{H_R} + y_{H_R}]/2$
$t_l = t_{H_R} - [z_{H_R} - y_{H_R}]/2C$
$z_k = z_l$
$t_k = t_l$
k = k + 1
Call Subroutine LARGE
$\hat{y} = f(t_{H_k})$
$\hat{y} = \hat{\phi}_{k-1}$
Yes
No
$\hat{\phi}_k = \hat{y}$
11
4
5

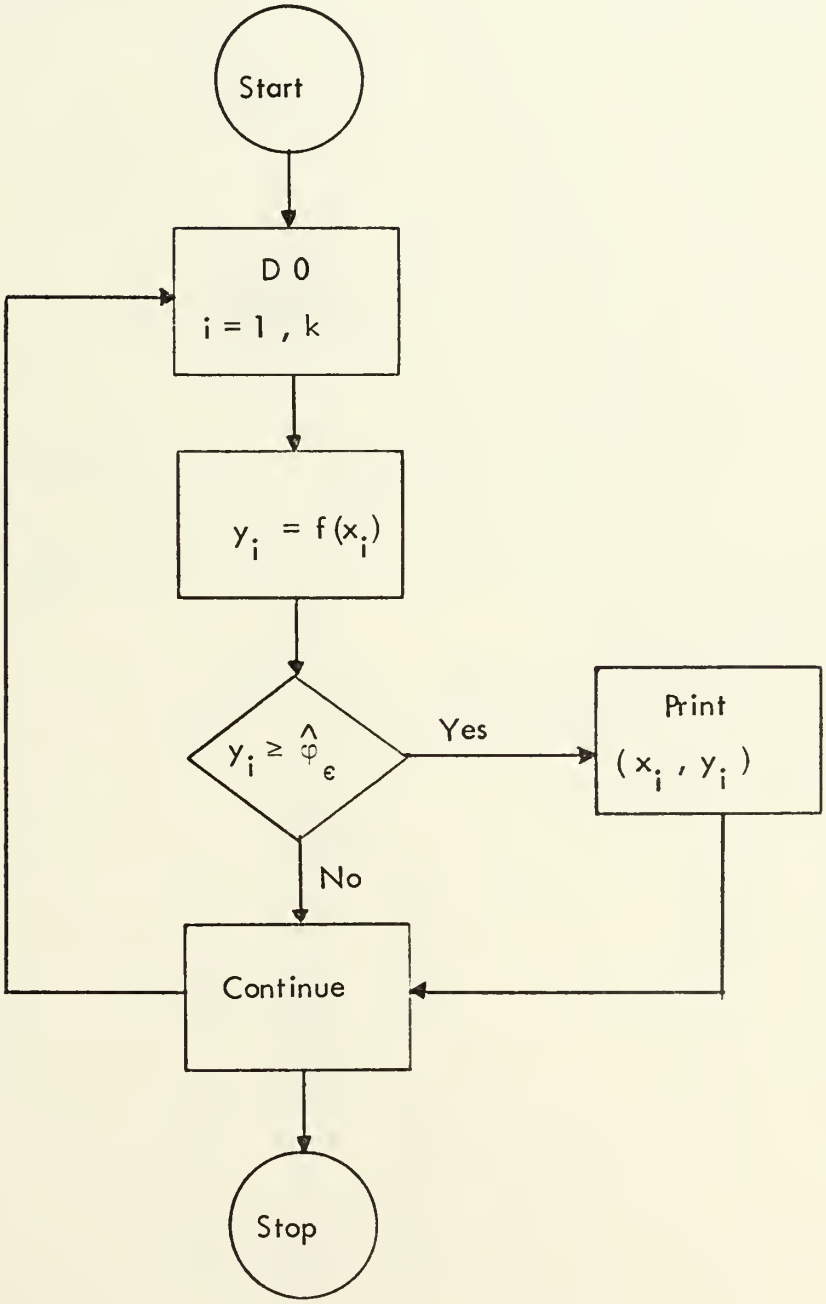

4
5
$\hat{\varphi}_k = \hat{\varphi}_{k-1}$
Reduction of Data
n = 0
DO 9
i = 1, k
$z_i < \hat{\varphi}_k$
Yes
Continue
No
n = n + 1
n = 1
Yes
No
8
6
7


6
7
$t_n = t_i$
$z_n = z_i$
index = k - i + 1
$t_n = t_i$
$z_n = z_i$
8
Continue
n ≠ index
Yes
No
k = Index
9
10


9
10
$t_{Hk} = t_k$
$y_{Hk} = f(t_k)$
$z_{Hk} = z_k$
11
No
$(z_{H_k} - \hat{\phi}_k) \leq \epsilon$
Yes
Print
Global max
$\hat{\phi}_k$
Go to location of
Global max Subprogram

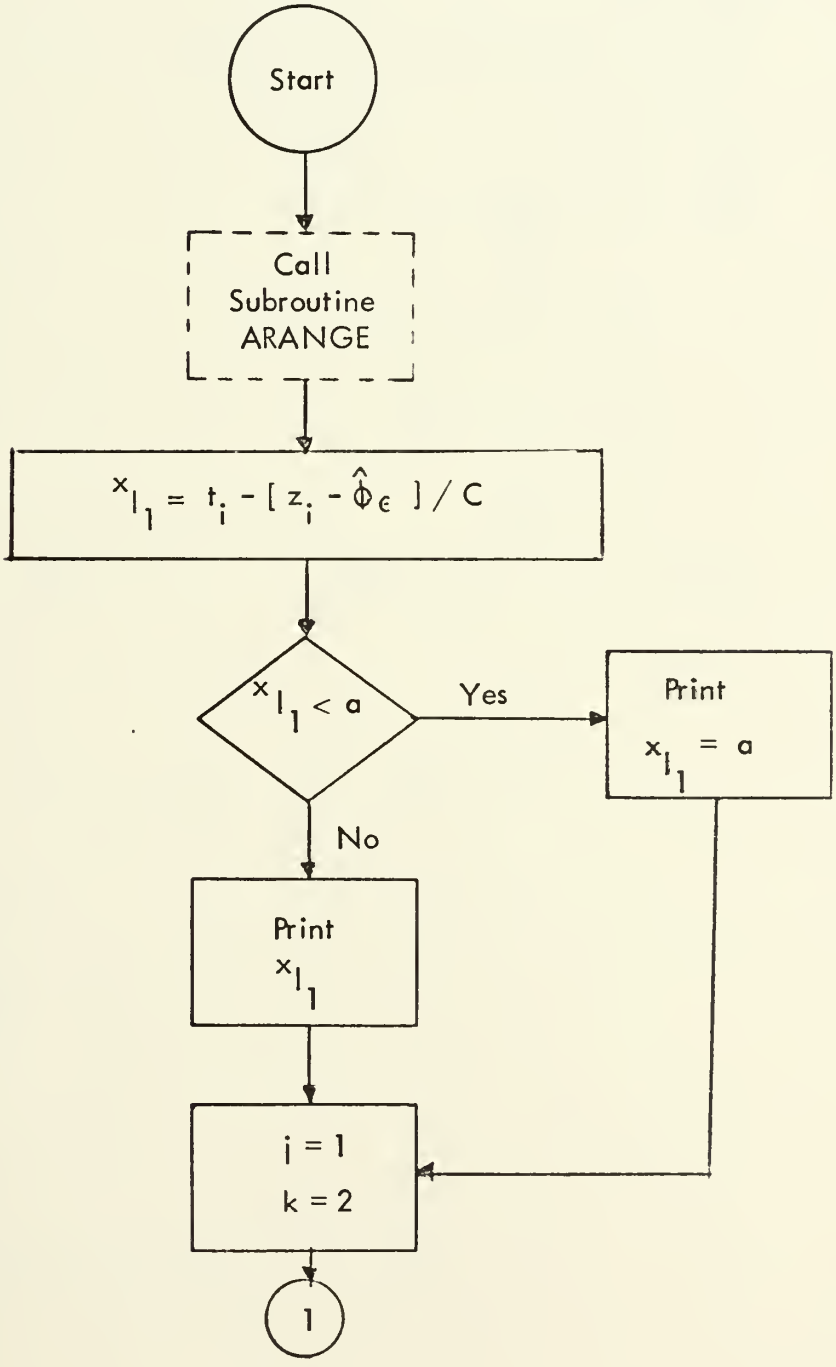
APPENDIX B

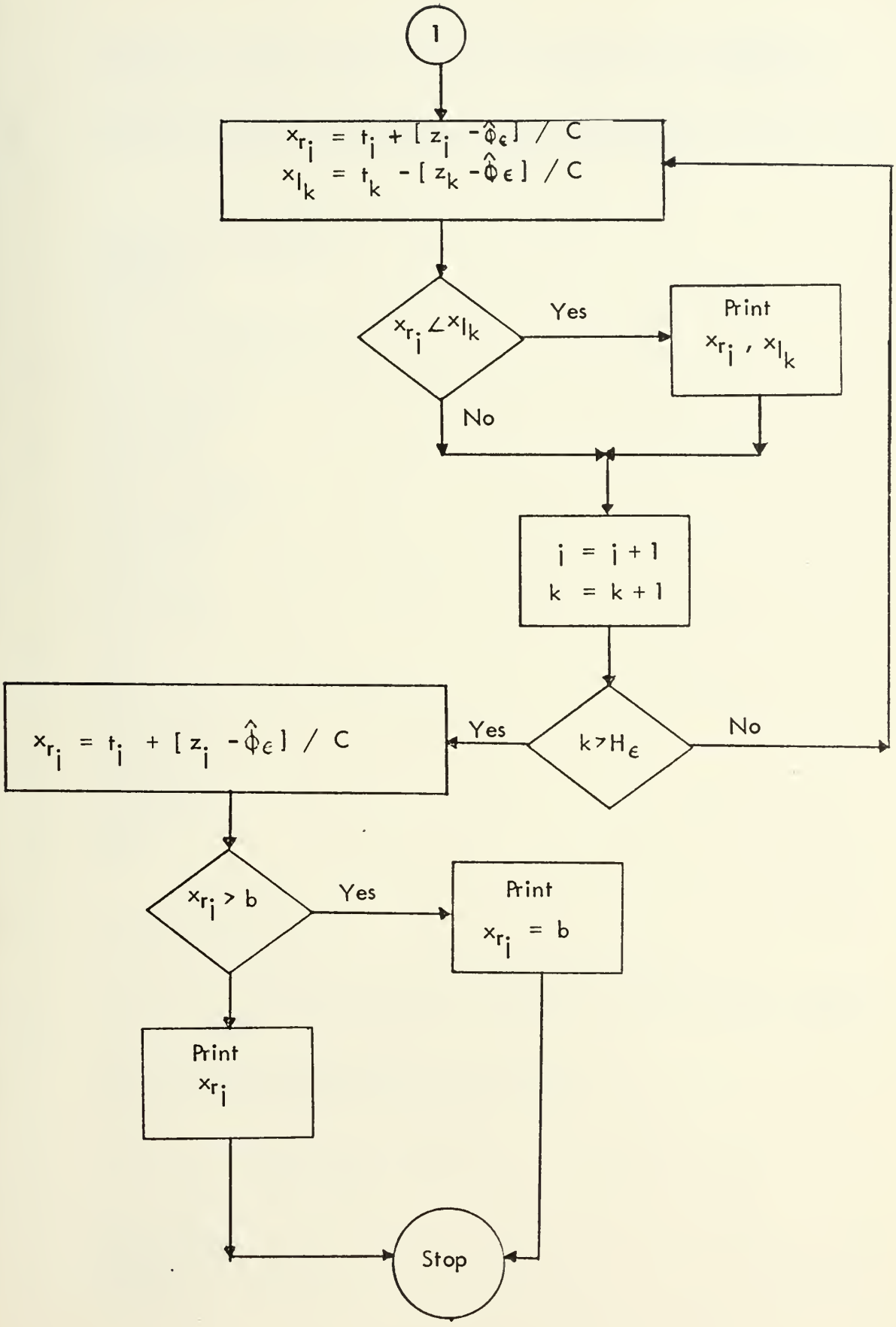
FLOW CHART FOR SUBPROGRAM ESTIMATING LOCATION OF GLOBAL MAXIMUM BY ALTERNATIVE II



APPENDIX C

FLOW CHART FOR SUBPROGRAM ESTIMATING LOCATION OF GLOBAL MAXIMUM BY ALTERNATIVE III




1
$x_{r_i} = t_i + [z_i - \hat{\Phi}_\epsilon] / C$
$x_{l_k} = t_k - [z_k - \hat{\Phi}\epsilon] / C$
$x_{r_i} \angle x_{l_k}$
Yes
Print
x_{r_i} , x_{l_k}
No
$i = i + 1$
$k = k + 1$
$k > H_\epsilon$
Yes
No
$x_{r_i} = t_i + [z_i - \hat{\Phi}_\epsilon] / C$
$x_{r_i} > b$
Yes
Print
$x_{r_i} = b$
Print
x_{r_i}
Stop

APPENDIX D

PROGRAM LISTING FOR FINDING THE GLOBAL MAXIMUM OF A DETERMINISTIC FUNCTION OF A SINGLE REAL VARIABLE

```
C       SUBROUTINE FOR REARRANGING THE VECTORS IN THE DATA SET
C       IN NONDECREASING ORDER OF SECOND COMPONENT


        SUBROUTINE LARGE(A,B,N)
        DIMENSION A,B,G,H
        X=A(N)
        Y=B(N-1)
        Z=B(N)
        J=1
        IF(N.EQ.2)GO TO 5
 1      IF(A(N).GE.A(J))GO TO 4
        L=N-2
        DO 2 I=J,L
        G(I)=A(I)
        H(I)=B(I)
 2      CONTINUE
        LA=J+2
        DO 3 M=LA,N
        A(M)=G(M-2)
        B(M)=H(M-2)
 3      CONTINUE
        A(J)=X
        B(J)=Y
        A(J+1)=X
        B(J+1)=Z
        GO TO 6
 4      J=J+1
        IF(J.GT.N)GO TO 6
        GO TO 1
 5      IF(A(N).GE.A(J))GO TO 6
        TEMP=A(J)
        A(J)=X
        A(N)=TEMP
        TEMP=B(J)
        B(J)=Z
        B(N)=TEMP
 6      RETURN
        END


C       MAIN PROGRAM

        DIMENSION X,Y,Z

        READ THE FUNCTION TO BE MAXIMIZED,F(XH)

C       FUNCTION SUBPROGRAMS FOR LOCATING THE TWO NEW PEAKS OF
C       F_K(XH) WHEN K IS GREATER THAN 2

        ZHI(ZH,YH)=(ZH+YH)/2.
        XHI1(XH,ZH,YH)=XH+(ZH-YH)/(2.*C)
        XHI2(XH,ZH,YH)=XH-(ZH-YH)/(2.*C)

        READ PARAMETERS A,B,C,EPS

C       ZERO(TH) ITERATION

        XI=(A+B)/2.
        YI=F(XI)
        PHIHAT=YI
        Z(2)=YI+C*(B-XI)
        X(1)=A
        X(2)=B
```

```
C       FIRST ITERATION

        XH=X(1)
        Y(1)=F(XH)
        ZH=Z(1)
        YH=Y(1)

C       ESTIMATE GLOBAL MAXIMUM

        IF(YH.GT.YI)GO TO 1
        PHIHAT=YI
        GO TO 2
 1      PHIHAT=YH

C       COMPUTE LOCATION OF NEW PEAK

 2      Z(3)=ZHI(ZH,YH)
        X(3)=XHI1(XH,ZH,YH)

C       DROP VECTOR FROM DATA SET WITH HIGHEST SECOND
C       COMPONENT AND ADD NEW VECTOR

        Z(1)=Z(3)
        X(1)=X(3)

C       SET K EQUAL TO THE NUMBER OF VECTORS IN THE DATA SET

        K=2

C       REARRANGE VECTORS IN DATA SET IN ORDER OF NON-
C       DECREASING SECOND COMPONENT

        CALL LARGE(Z,X,K)

C       SECOND ITERATION

        XH=X(2)
        Y(2)=F(XH)
        ZH=Z(2)
        YH=Y(2)

C       ESTIMATE GLOBAL MAXIMUM

        IF(YH.GE.PHIHAT)GO TO 3
        GO TO 4
 3      PHIHAT=YH

C       COMPUTE LOCATION NEW PEAK

 4      Z(3)=ZHI(ZH,YH)
        X(3)=XHI2(XH,ZH,YH)

C       DROP VECTOR FROM DATA SET WITH HIGHEST SECOND
C       COMPONENT AND ADD NEW VECTOR

        Z(2)=Z(3)
        X(2)=X(3)

C       SET K EQUAL TO THE NUMBER OF VECTORS IN THE DATA SET

        K=2

C       REARRANGE VECTORS IN DATA SET IN ORDER OF NON-
C       DECREASING SECOND COMPONENT

        CALL LARGE(Z,X,K)

C       K(TH) ITERATION

C       COMPUTE LOCATION OF TWO NEW PEAKS
```

```
 5      Z(K+1)=ZHI(ZH,YH)
        X(K+1)=XHI1(XH,ZH,YH)
        Z(K+2)=ZHI(ZH,YH)
        X(K+2)=XHI2(XH,ZH,YH)

C       DROP VECTOR FROM DATA SET WITH HIGHEST SECOND
C       COMPONENT AND ADD NEW VECTOR

        Z(K)=Z(K+2)
        X(K)=X(K+2)

C       SET K EQUAL TO THE NUMBER OF VECTORS IN THE DATA SET

        K=K+1

C       REARRANGE VECTORS IN DATA SET IN ORDER OF NON-
C       DECREASING SECOND COMPONENT

        CALL LARGE(Z,X,K)

C       SET YHAT EQUAL TO THE SAMPLE VALUE FOR THE ABSCISSA
C       OF THE HIGHEST PEAK IN THE DATA SET

        YHAT=F(X(K))

C       ESTIMATE GLOBAL MAXIMUM

        IF(YHAT.GE.PHIHAT)GO TO 6
        GO TO 7
 6      PHIHAT=YHAT

C       REDUCTION OF DATA

 7      N=0
        DO9 I=1,K
        IF(Z(I).LT.PHIHAT)GO TO 9
        N=N+1
        IF(N.EQ.1)GO TO 8
        X(N)=X(I)
        Z(N)=Z(I)
        GO TO 9
 8      INDEX=K-I+1
        X(N)=X(I)
        Z(N)=Z(I)
 9      CONTINUE

C       DETERMINE THE NUMBER OF VECTORS LEFT IN THE DATA SET

        IF(N.NE.INDEX)GO TO 10
        K=INDEX

C       BEGIN (K+1)ST SAMPLE

 10     XH=X(K)
        Y(K)=F(XH)
        ZH=Z(K)
        YH=Y(K)

C       STOPPING RULE

        IF((Z(K)-PHIHAT).LE.EPS)GO TO 11
        GO TO 5
 11     PRINT,PHIHAT

C       GO TO LOCATION OF GLOBAL MAX SUBPROGRAM
```

APPENDIX E

SUBPROGRAM LISTING FOR ESTIMATING THE LOCATION OF THE GLOBAL MAXIMUM BY ALTERNATIVE II

```
C       EVALUATE THE FUNCTION FOR EACH FIRST COMPONENT OF DATA
C       SET D_ε

        DOJ=1,K
        Y(J)=F(X(J))

C       LOCATE THE COORDINATES FOR WHICH THE VALUES OF THE
C       FUNCTION ARE GREATER THAN OR EQUAL TO THE ESTIMATED
C       MAXIMUM OBTAINED FROM THE MAIN PROGRAM

        IF(Y(J).GE.PHIHAT)GO TO 13
        GO TO 15
  13    PRINT (X(J),Y(J))
  14    FORMAT
  15    CONTINUE
        STOP
```

APPENDIX F

SUBPROGRAM LISTING FOR ESTIMATING THE LOCATION OF THE GLOBAL MAXIMUM BY ALTERNATIVE III

```
C       SUBROUTINE FOR REARRANGING ALL VECTORS IN DATA SET Dε
C       IN NONDECREASING ORDER OF FIRST COMPONENT


        SUBROUTINE ARANGE(A,B,N)
        DIMENSION A(N),B(N)
        L=N-1
        DO 10 J=1,L
        JP1=J+1
        DO 10 I=JP1,N
        IF(A(J).LE.A(I))GOTO10
        TEMP=A(J)
        A(J)=A(I)
        A(I)=TEMP
        TEMP=B(J)
        B(J)=B(I)
        B(I)=TEMP
 10     CONTINUE
        RETURN
        END

C       REARRANGE DATA IN Dε IN ORDER OF NONDECREASING FIRST
C       COMPONENT

 30     CALL ARRANGE(X,Z,K)

C       DETERMINE THE LEFTMOST ENDPOINT OF THE GENUINE
C       UNCERTAINTY SET V

        XL(1)=X(1)-(Z(1)-PHIHAT)/C
        IF(XL(1).LT.A)GO TO 31
        GO TO 32
 31     XL(1)=A
 32     PRINT XL(1)
 00     FORMAT
        J=1
        L=1

C       DETERMINE THE RIGHT AND LEFT ENDPOINTS OF REMAINING
C       INTERVALS IN THE SET V

 34     XR(J)=X(J)+Z(J)-PHIHAT)/C
        XL(L)=X(L)-(Z(L)-PHIHAT)/C
        IF(XR(J).LT.XL(L))GO TO 35
        GO TO 37
 35     PRINT, XR(J),XL(J)
 36     FORMAT
 37     J=J+1
        L=L+1
        IF(L.GT.K)GO TO 38
        GO TO 34

C       DETERMINE RIGHTMOST ENDPOINT OF V

 38     XR(J)=X(J)+(Z(J)-PHIHAT)/C
        IF(XR(J).GT.B)GO TO 39
        GO TO 40
 39     XR(J)=B
 40     PRINT, XR(J)
 41     FORMAT
        STOP
```

BIBLIOGRAPHY

1. Berman, Gerald, "Minimization By Successive Approximation, " J. SIAM Numer. Anal . v. 3, No. 1, p. 123-133, 1966.

2. Brooks, S. H., "A Comparison of Maximum-Seeking Methods," J. Operations Res. Soc., v. 7, p. 430-457, 1959.

3. Brooks, S. H., "A Discussion of Random Methods of Seeking Maxima," J. Operations Res. Soc., v. 6, p. 244-251, 1958.

4. Hill, D. J., "A Search Technique For Multimodal Surfaces," IEEE Transactions on Systems Science and Cybernetics, v. ssc -5, No. 1, p. 1-8, January 1969.

5. Institute of Statistics, Texas A & M University Report 18, Statistical Control for Nonlinear Optimization, by H. O. Hartley and R. C. Pfaffenberger, September 1969.

6. Karnopp, D. C., "Random Search Techniques for Optimization Problems," Automatica, v. 1, p. 111-121, 1963.

7. Kiefer, J., "Optimum Sequential Search and Approximation Methods Under Minimum Regularity Assumptions," J. Soc. Indust. Appl. Math., v. 5, No. 3, p. 105-136, September 1957.

8. Matyas, J., "Random Optimization," Automation and Remote Control, v. 26, p. 244-251, 1965.

9. Pijavskii, S. A., "An Algorithm For Finding the Absolute Minimum of Functions," Theory of Optimal Solutions, No. 2, Akad. Nank, Ukrain, SSR, Kiev, 1967.

10. Shubert, B. O., "A Sequential Method Seeking the Global Maximum of a Function, " to appear.

11. Spang, H. A., "A Review of Minimization Techniques for Nonlinear Functions," SIAM Review, v. 4, No. 4, p. 343-365, October 1962.

12. Vaysbord, E. M. and Yudin, D. B., "Multiextremal Stochastic Approximation," Engineering Cybernetics, No. 5, p. 1-10, 1968.

13. Wilde, D. J., Optimum Seeking Methods, p. 1-51, Prentice-Hall, Inc., 1964.

14. Zachharov, V. V., "A Random Search Method," Engineering Cybernetics, No. 2, p. 26-30, 1969.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Asst Professor B. O. Shubert, Code 55 Sy Department of Operations Analysis Naval Postgraduate School Monterey, California 93940	1
4.	CAPT Ray Lovell Springfield, USMCR R.D. # 2 Wellsburg, New York 14894	1
5.	Department of Operations Analysis Naval Postgraduate School Monterey, California 93940	1

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)

Naval Postgraduate School
Monterey, California 93940

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

3. REPORT TITLE

A Computerized Algorithm for Sequential Search of the Global Maximum

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Master's Thesis, September 1970

5. AUTHOR(S) (First name, middle initial, last name)

Ray L. Springfield

6. REPORT DATE

September 1970

7a. TOTAL NO. OF PAGES

104

7b. NO. OF REFS

14

8a. CONTRACT OR GRANT NO.

8b. PROJECT NO.

8c.

8d.

9a. ORIGINATOR'S REPORT NUMBER(S)

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY

Naval Postgraduate School
Monterey, California 93940

13. ABSTRACT

A sequential search procedure for maximization of a single variable multimodal objective function is designed and investigated in this research. Existing sequential procedures require the function to be unimodal. Nonsequential methods, though not restricted in this sense, require a large number of samples. Results show that the proposed sequential method is in this case preferable.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811

Security Classification

A-31408

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
sequential search						
computerized algorithm						
global maximum						
maximization procedure						
minimization procedure						
sampling procedure						
sequential sampling						
sample rule						
Lipschitzian functions						